ANSCHUETZ/WEISGERBER/ANSCHUETZ

GAME DEVELOPMENT NOTES

Written by Robert & Eric Anschuetz and John Weisgerber

Version 1.0

August 22, 2017

CHANGE NOTES

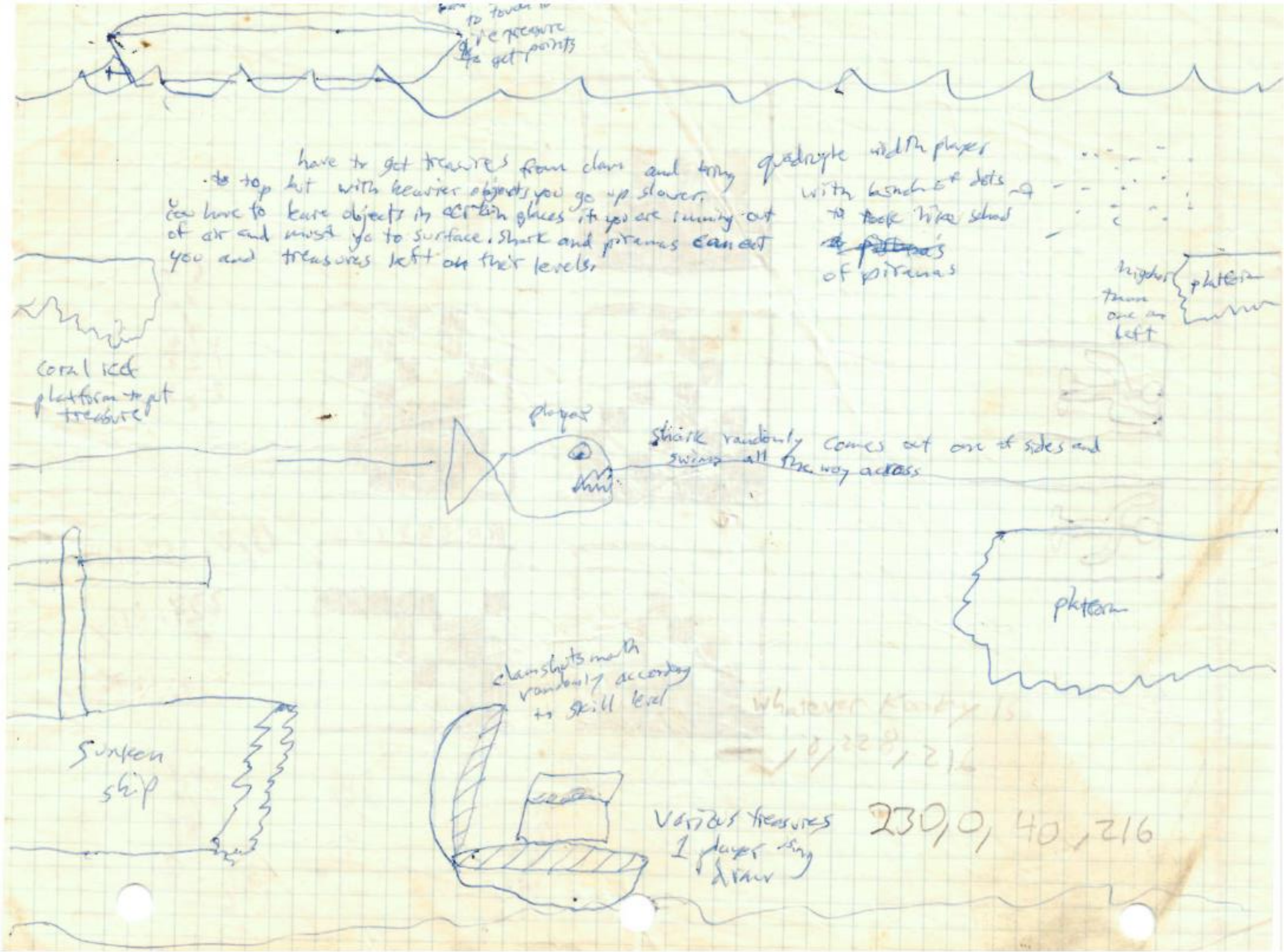
Version 1.0 (August 22, 2017)

- Initial Release

TABLE OF CONTENTS

A/W/A GAME DEVELOPMENT NOTES

KOOKY DIVER



have to get treasures from clam and bring to top but with heavier objects you go up slower. You have to leave objects in certain places if you are running out of air and must go to surface. Shark and piranas can eat you and treasures left on their levels.

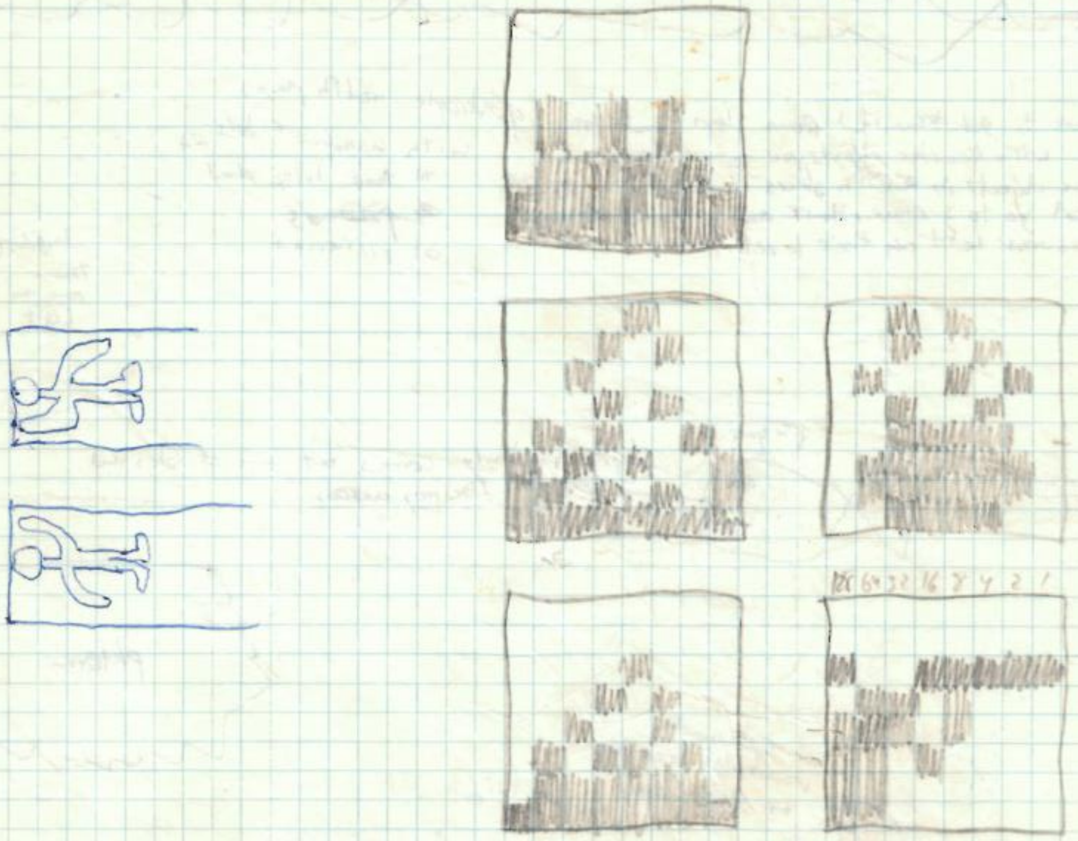
quadruple width player with bunch of dots to look like school ~~of pattern~~ of piranas

shark randomly comes out one of sides and swims all the way across

clam shuts mouth randomly according to skill level

various treasures 1 player sing drawn

2



64
32
96
128
224

0, 0, 159, 104,

234, 218,

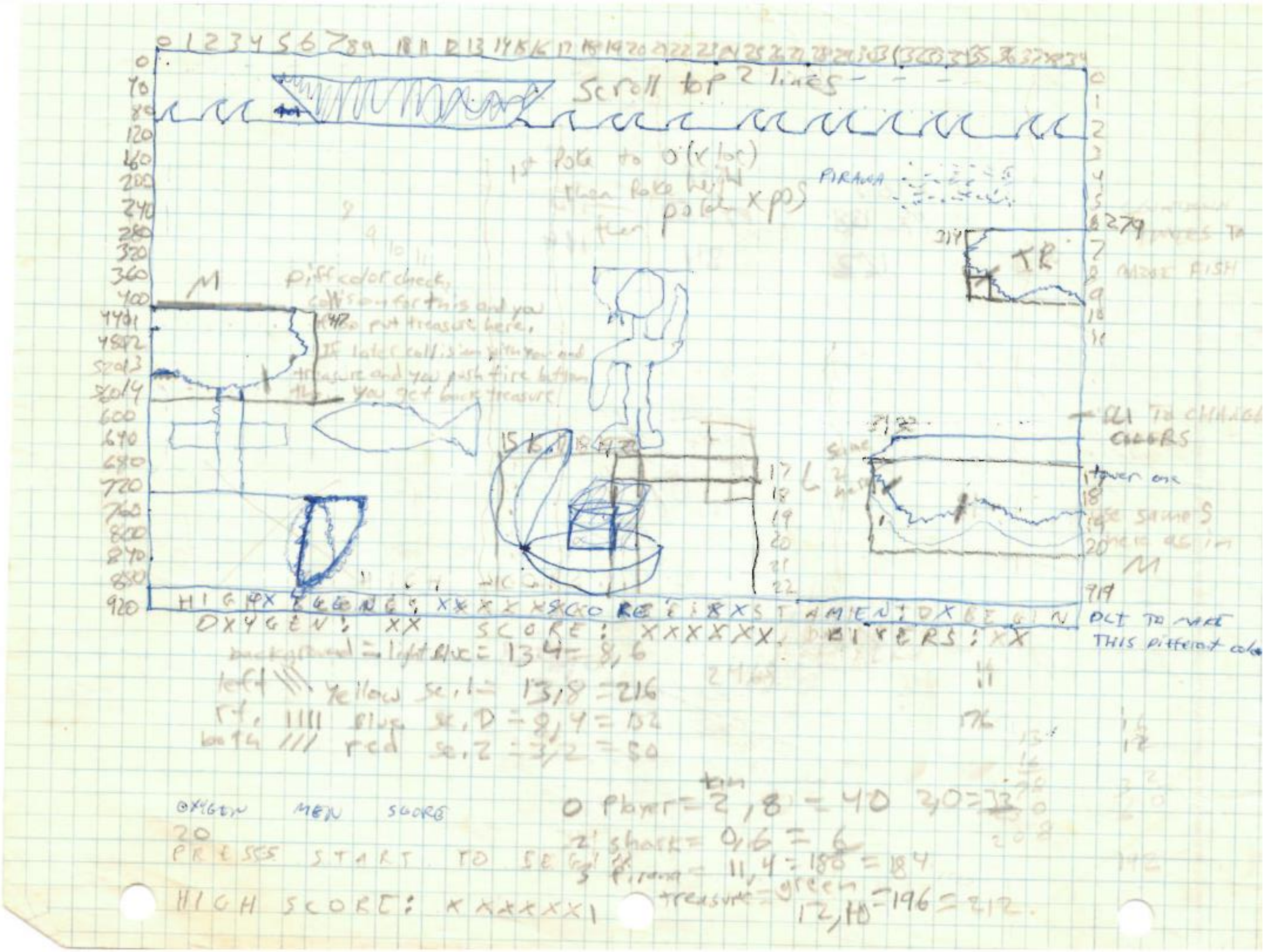
192, 192,

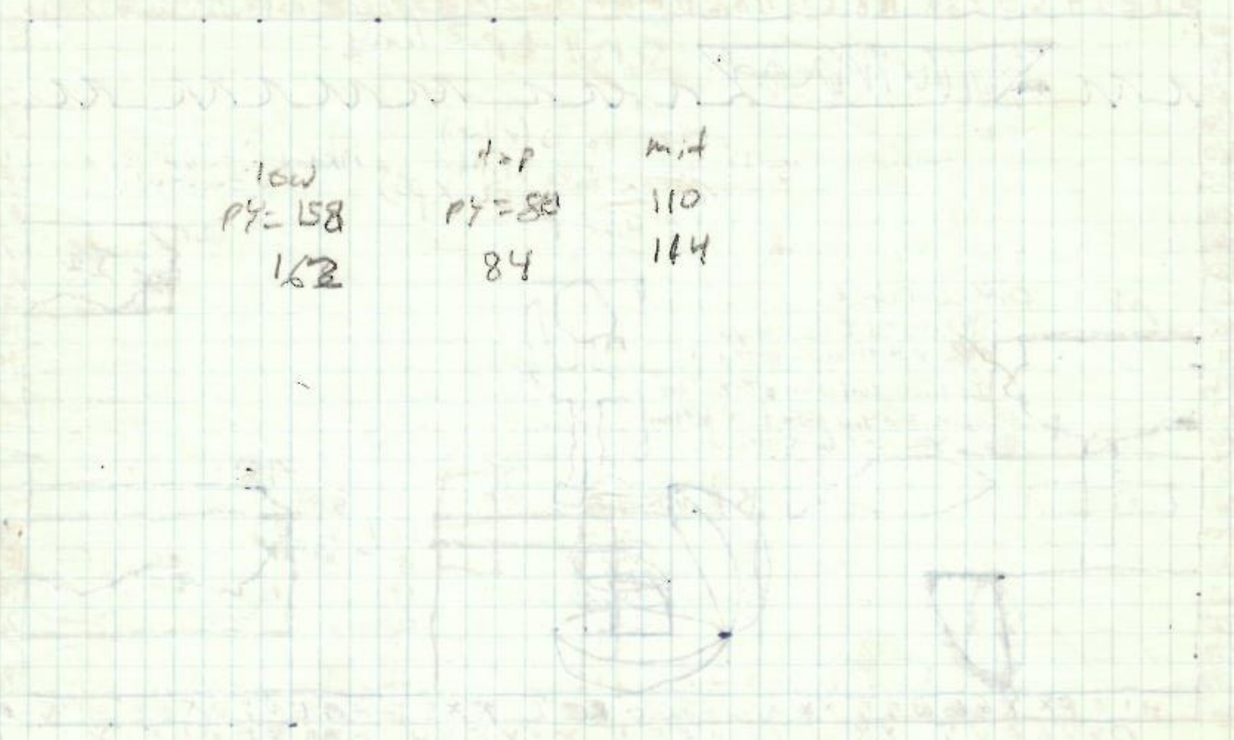
128
64
192
16

96
159

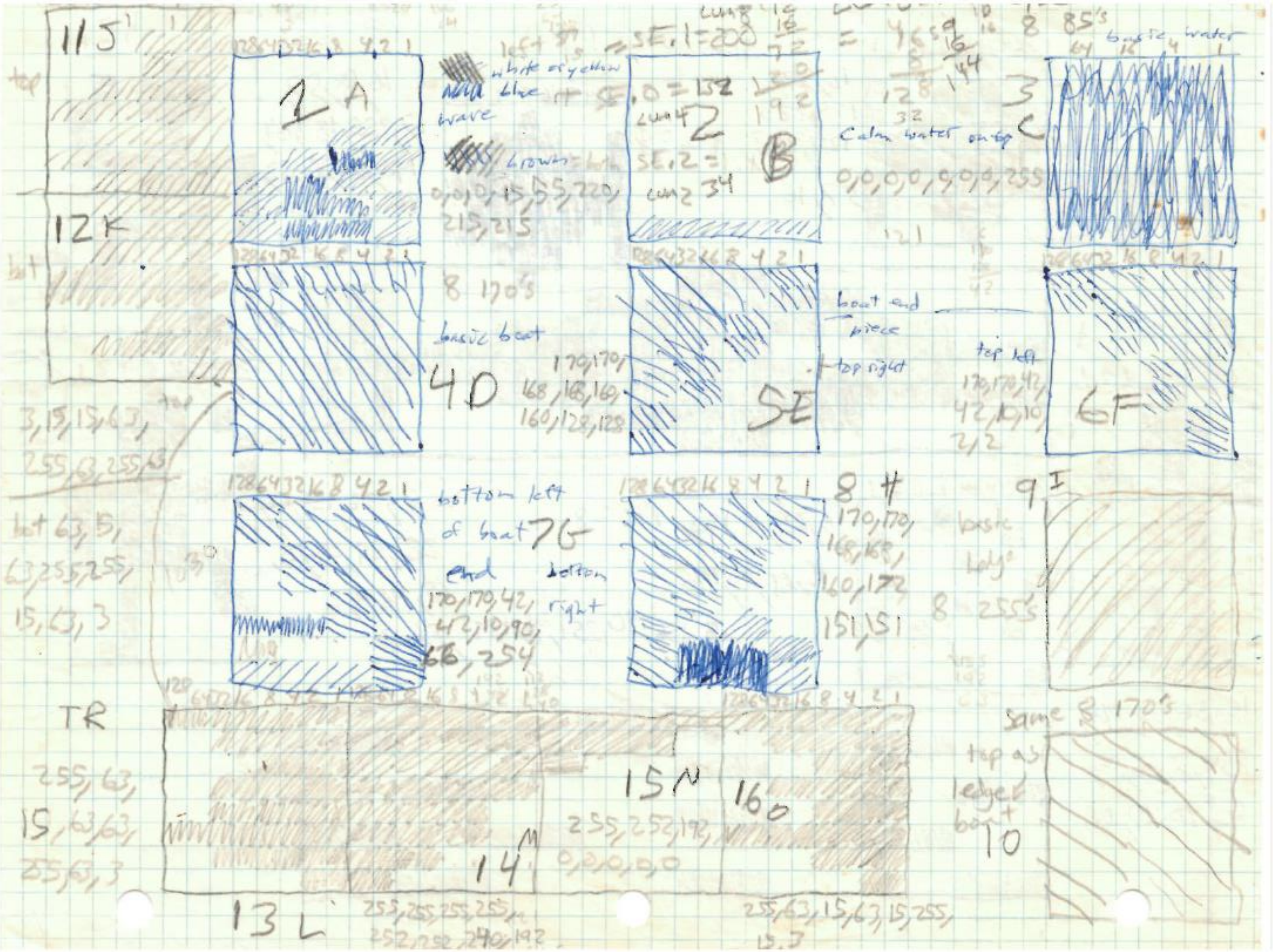
104
128
264
2

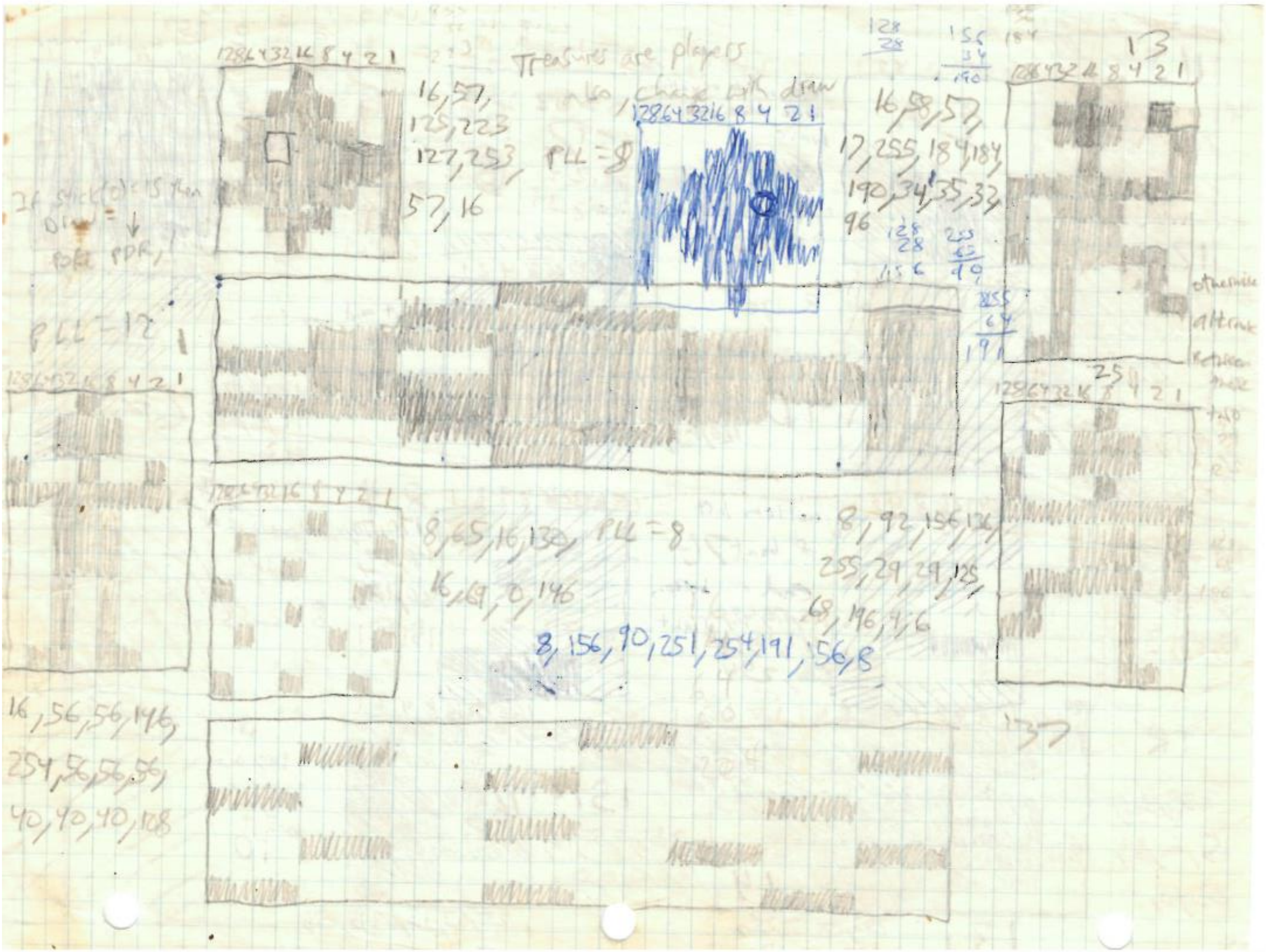
192
16

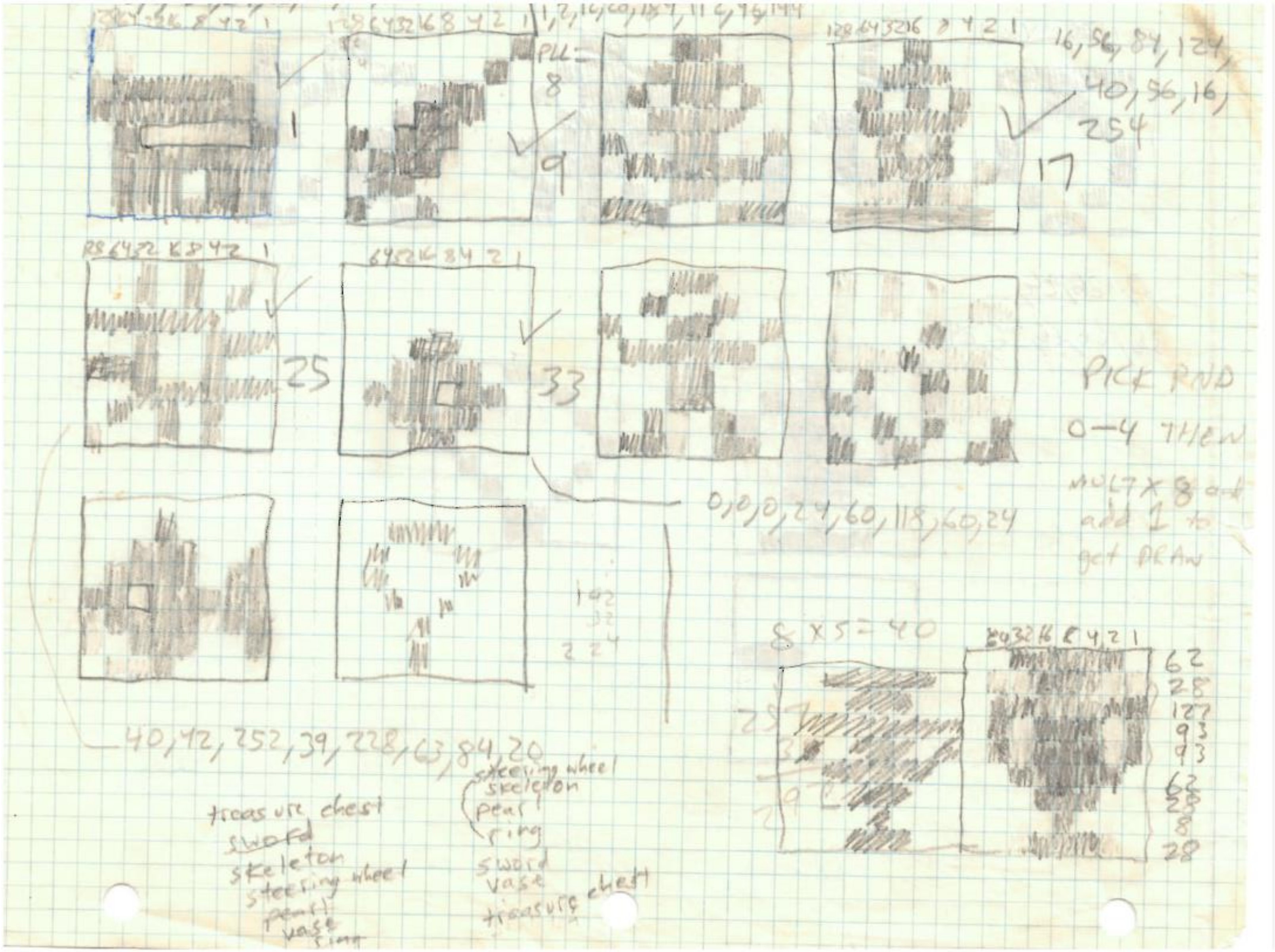

Scroll top 2 lines
1st Poke to 0 (x loc)
then Poke with
pole x pos
then
PIRANA
279
219
TR
MORE FISH
M
Diff color check,
collision for this and you
put treasure here,
If later collision with you and
treasure and you push fire button
then you get back treasure.
DLI TO CHANGE COLORS
tower one
use same S
same as in M
919
HIGH X OXYGEN: XX SCORE: XXXXXXX AMIEN: OXYGEN
DLI TO MAKE THIS different color
OXYGEN: XX SCORE: XXXXXX, DIVERS: XX
background = light blue = 13,4 = 8,6
left \\\ yellow se, 1 = 13,8 = 216
rt. |||| blue se, D = 8,4 = 132
both /// red se, 2 = 3,2 = 50
OXYGEN MEN SCORE
20
PRESSES START TO SE
HIGH SCORE: XXXXXXX1
0 Player = 2,8 = 40 30 = 33
2 shark = 0,6 = 6
3 Pirana = 11,4 = 180 = 184
treasure = green 12,10 = 196 = 212



low	top	mid
PY= 158	PY=80	110
162	84	144


11 J
12 K
top
bot
1 A
128 64 32 16 8 4 2 1
white or yellow
blue
wave
brown
0,0,0,15,55,220,
215,215
SE,1=200
SE,0=132
cum 4
2
B
SE,2 =
cum 2 34
Calm water on top
3
C
0,0,0,0,9,0,0,255
8 85's
basic water
8 170's
basic boat
4 D
170,170,
168,168,168,
160,128,128
5 E
boat end
piece
top right
top left
170,170,42,
42,10,10,
2,2
6 F
3,15,15,63,
255,63,255,63
bot 63,15,
63,255,255,
15,63,3
bottom left
of boat
7 G
end bottom
right
170,170,42,
42,10,90,
66,254
8 H
170,170,
168,168,
160,172
151,151
9 I
basic
ledge
8 255's
same 8 170's
top as
ledge
bot
10
TR
255,63,
15,63,63,
255,63,3
13 L
14 M
15 N
16 O
255,252,192,
0,0,0,0,0
255,255,255,255,
252,252,240,192
255,63,15,63,15,255,
15,3


128 64 32 16 8 4 2 1
16, 57,
125, 223
127, 253,
57, 16
Treasures are players
also, change with draw
PLL = 8
128 64 32 16 8 4 2 1
128
28
156
34
190
184
16, 58, 57,
17, 255, 184, 184,
190, 34, 35, 32,
96
128
28
156
255
65
190
255
64
191
13
128 64 32 16 8 4 2 1
otherwise
attract
retreat mode
25
128 64 32 16 8 4 2 1
140
If stick(0)=15 then
dir = ↓
poke ppr,
PLL = 12
128 64 32 16 8 4 2 1
128 64 32 16 8 4 2 1
8, 65, 16, 130, PLL = 8
16, 69, 0, 146
8, 92, 156, 136,
255, 29, 29, 125,
68, 196, 4, 6
8, 156, 90, 251, 254, 191, 56, 8
16, 56, 56, 146,
254, 56, 56, 56,
40, 40, 40, 108
137



1,2,16,60,184,116,96,144

16,56,84,124,40,56,16,254

0,0,0,24,60,118,60,24

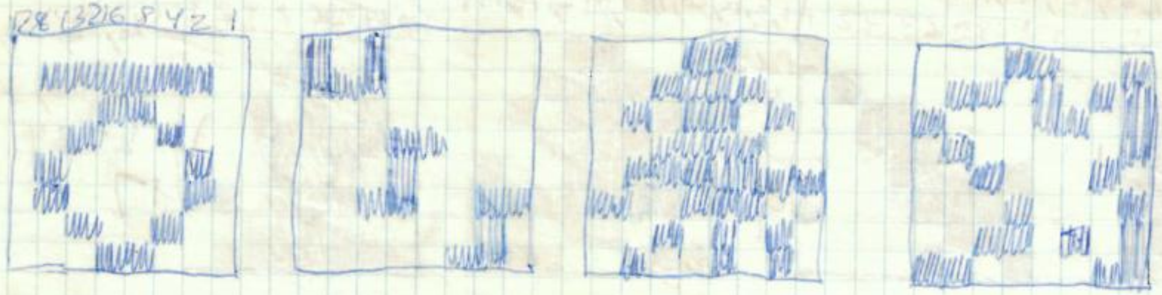
40,42,252,39,228,63,84,20

PICK RND 0–4 THEN MULT X 8 and add 1 to get DRAW

8 X 5 = 40

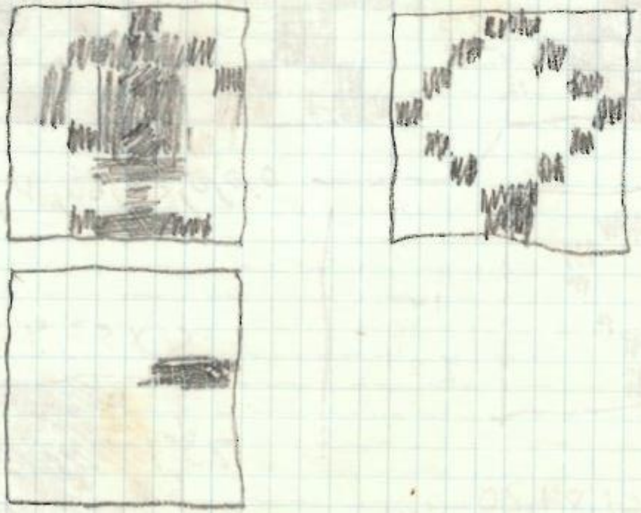
treasure chest
~~sword~~
skeleton
steering wheel
pearl
vase
ring

steering wheel
skeleton
pearl
ring
sword
vase
treasure chest



0,126,24,

36,66,66,36,24



3 = 87
243 = 247 128 64 | 32 16 | 8 4 | 2 1
48 = 117 58 = 122
192 = 213
240 = 245
15 = 95
60 = 125
204 = 221
14 = 94

255, 255, 170, 170, 170, 170, 170, 170

Outline is Black ▨ = 00

main color is Dk. Blu ▨ = 8×16 +4 132

shadow is Lt. Blu ▨ = 9×16 +6 = 150

29 ! 3, 3, 14, 14, 58, 58, 234, 234
87 94 122 234

30 " 8 234's

31 # 234, 234, 58, 58, 14, 14, 3, 3
same

32 $ 234, 234, 58, 58, 14, 14, 3, 3

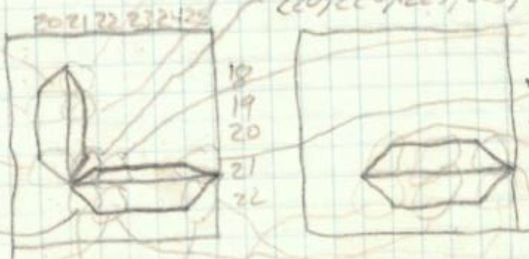
33 ' 170, 170, 170, 170, 170, 170, 255, 255

34 % 171, 171, 172, 172, 176, 176, 192, 192
173

192, 192, 240, 240, 220, 220, 220, 220 35 &
221

8 220's 36 *

220, 220, 223, 223, 245, 245, 213, 255 37 '



38 (0, 0, 255, 255, 85, 85, 85, 255

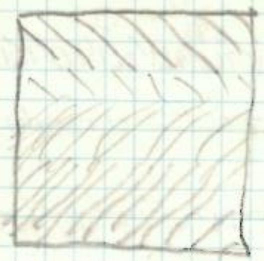
39) 0, 0, 0, 192, 240, 124, 95, 255
125

40 [3, 3, 14, 14, 58, 58, 234, 255

192, 192,

41] 176, 176, 172, 172, 171, 255

shadow on box



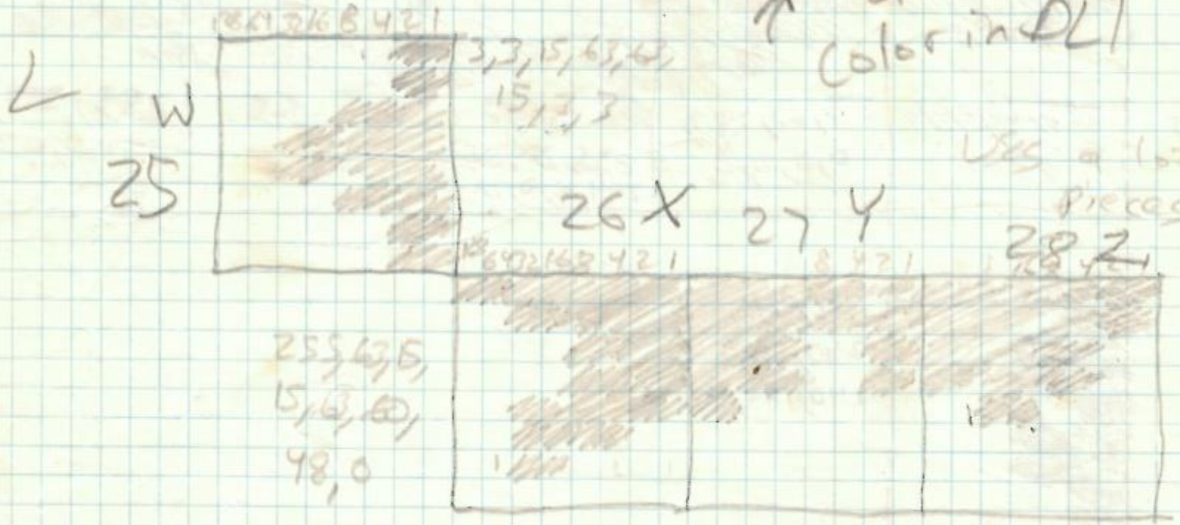
=

39

170, 170, 170

255, 255, 255, 255, 255

also change color in DL!



3,3,15,63,63,
15,3,3

Uses a lot same
pieces

255,63,15,
15,63,60,
48,0

255,255,243,243,
192,0,0,0

255,255,252,
252,48,0,0,0

230

63=127

15=95

Tape 13

150-171

175-202

210-240
250-280

290-320

330-391

150-197

250-298

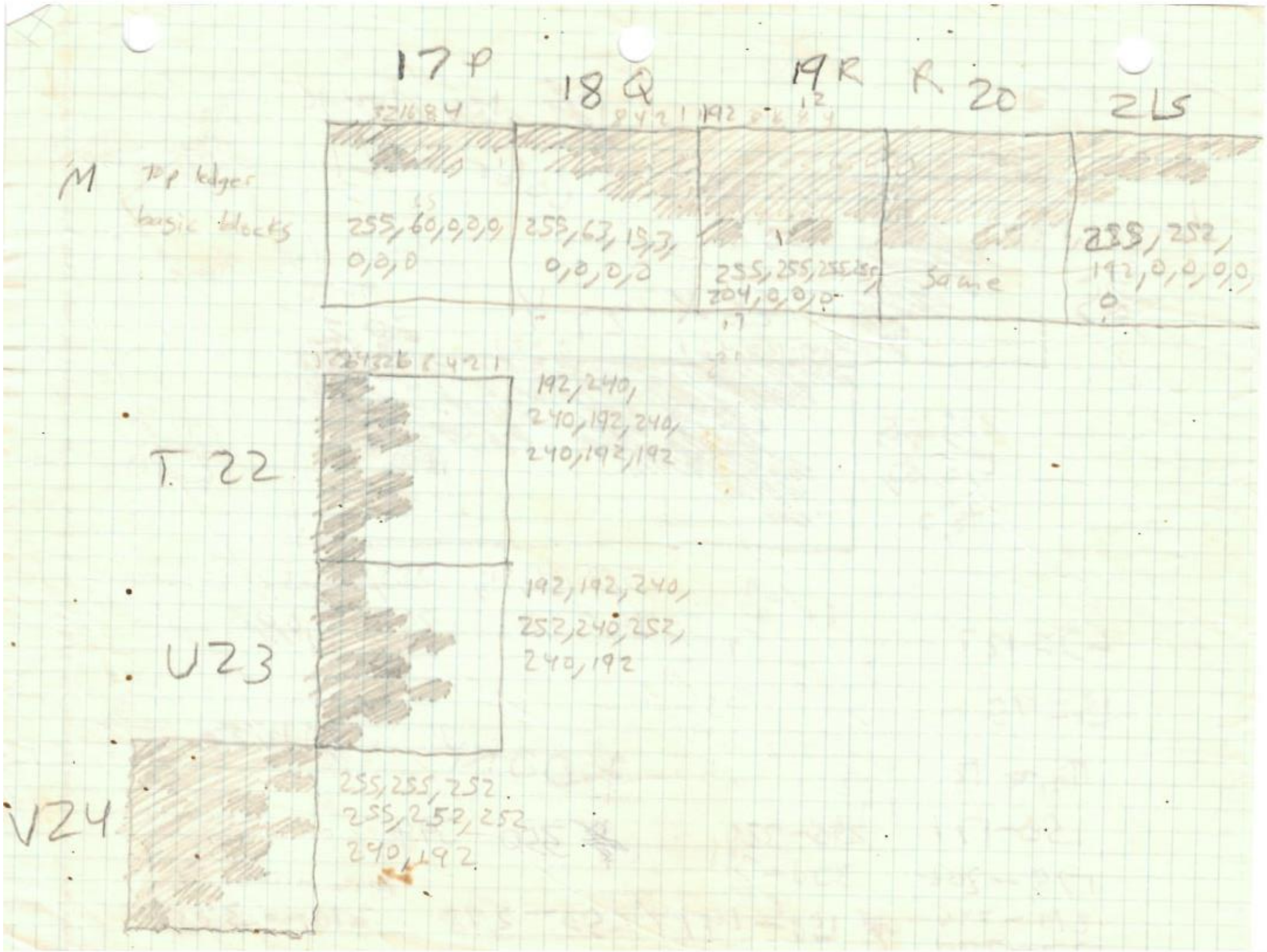
KOOKY

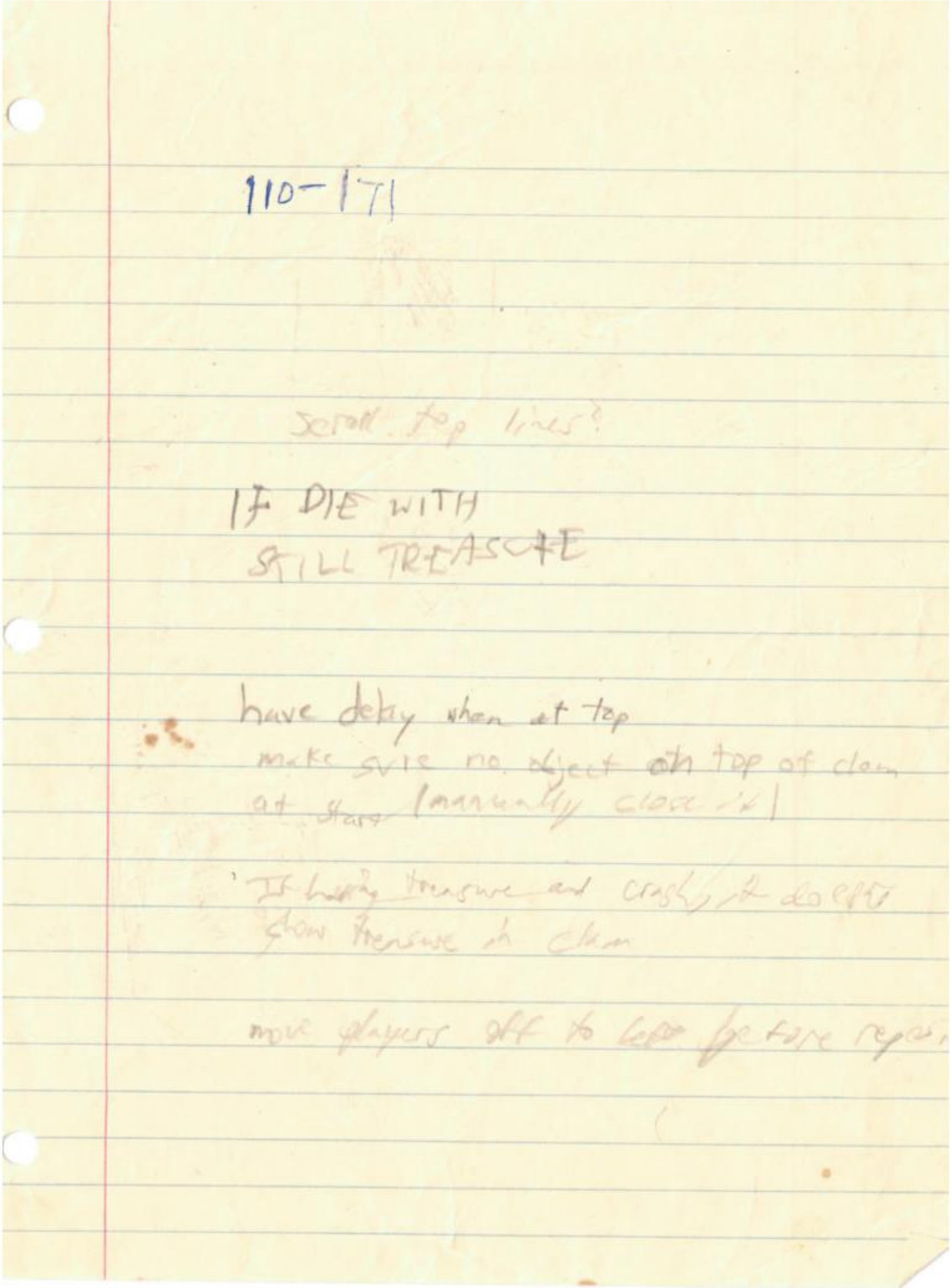
350-394

DIVER

Alien assailants

310-338


17 P
18 Q
19 R
R 20
2 S
32 16 8 4
8 4 2 1
192
M
Top Edges
basic blocks
255, 60, 0, 0, 0
0, 0, 0
255, 63, 15, 3,
0, 0, 0, 0
255, 255, 255, 255,
204, 0, 0, 0
Same
255, 252,
192, 0, 0, 0, 0,
0
128 64 32 16 8 4 2 1
T 22
192, 240,
240, 192, 240,
240, 192, 192
U 23
192, 192, 240,
252, 240, 252,
240, 192
V 24
255, 255, 252
255, 252, 252
240, 192



110-171

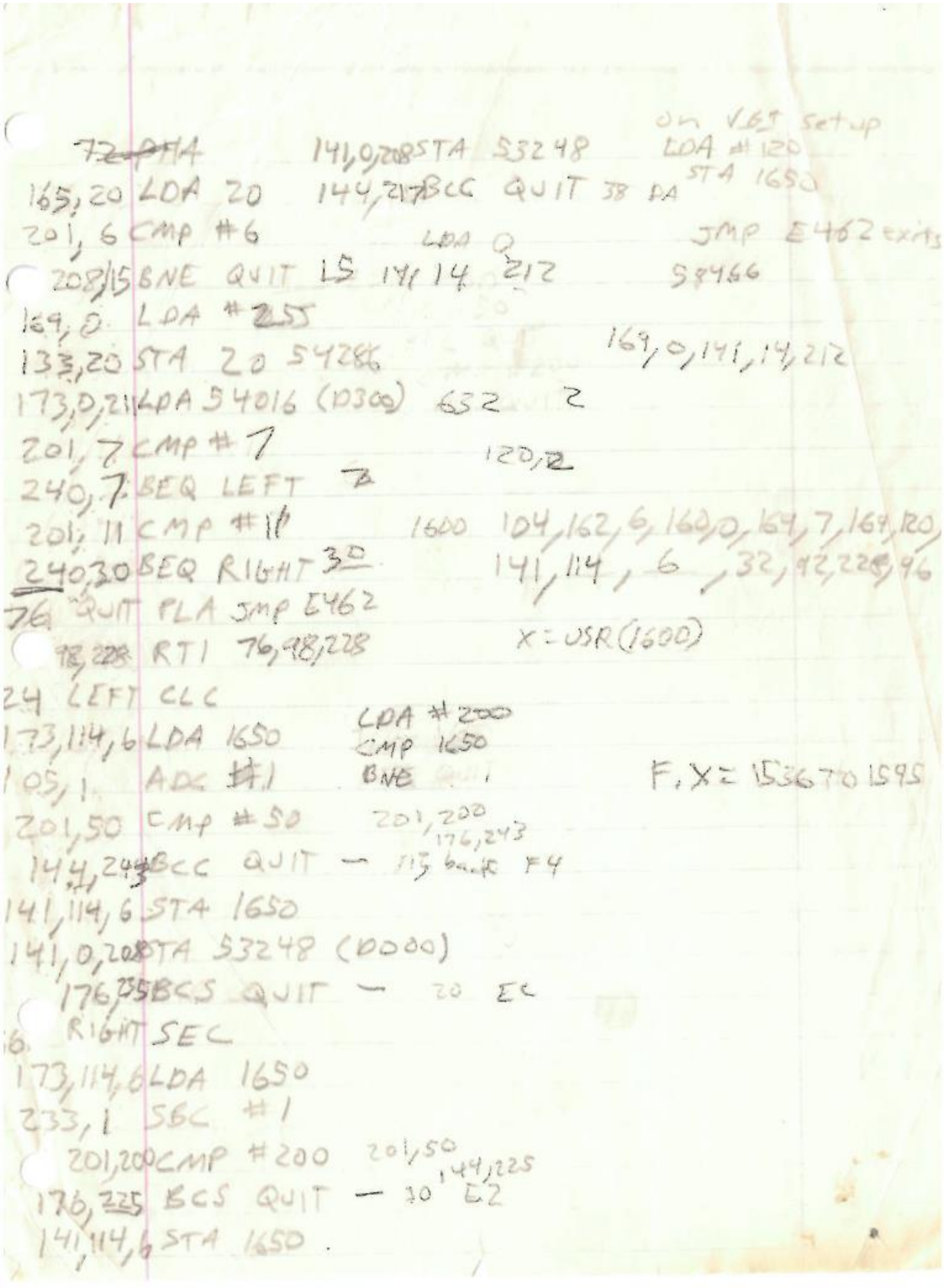
scroll top lines?

IF DIE WITH
STILL TREASURE

have delay when at top
make sure no object on top of clam
at start (manually close it)

If having treasure and crash, it doesn't
show treasure in clam

move players off to left before repai



```
                                            on VBI setup
~~72 PHA~~         141,0,208STA 53248       LDA #120
165,20 LDA 20      144,217BCC QUIT 38 DA    STA 1650
201, 6 CMP #6                               JMP E462 exits
                            LDA Q
208,15 BNE QUIT  15  141,14  212            58466
169, 0  LDA #255
133,20 STA 20 54286                     169,0,141,14,212
173,0,211LDA 54016 (D300)  652     2
201, 7 CMP #7                    120,2
240, 7 BEQ LEFT     2
201, 11 CMP #11         1600  104,162,6,160,0,169,7,169,120,
240,30 BEQ RIGHT 30           141,114, 6 ,32,92,228,96
76  QUIT PLA JMP E462
 98,228 RTI  76,98,228               X=USR(1600)
24  LEFT CLC
                          LDA #200
173,114,6 LDA 1650        CMP 1650
105,1   ADC #1            BNE QUIT              F.X= 1536 TO 1595
201,50  CMP #50           201,200
                              176,243
144,243BCC  QUIT  —  115 back F4
141,114,6 STA 1650
141,0,208STA 53248 (D000)
   176,235BCS QUIT  —    20  EC
56  RIGHT SEC
173,114,6LDA  1650
233,1  SBC  #1
  201,200CMP #200   201,50
                        144,225
 176,225 BCS  QUIT  —  30  E2
 141,114,6 STA 1650 .
```

POKE 1539,skill

moves player 0 left and right according to joystick

```
FOR X=1536 TO 1633:READ A:POKE X,A:NEXT X
DATA 165,20,201,2,208,15,169,0,133,20,173,
120,2,201,7,240,7,201,11,240,30,76,98,228,169,
186,205,114,6,208,8
DATA 169,185,141,114,6,24,144,238,238,114,6,173,
114,6,141,0,208,24,144,226,169,52,205,114,6,208,
8,164,53
DATA 141,114,6,24,144,211,206,114,6,173,114,6,141,0,
208,24,144,199
DATA 104,162,6,160,0,169,7,32,92,228,169,121,141,
114,6,169,0,133,20,96
X=USR(1614)
```

skill

rtmost

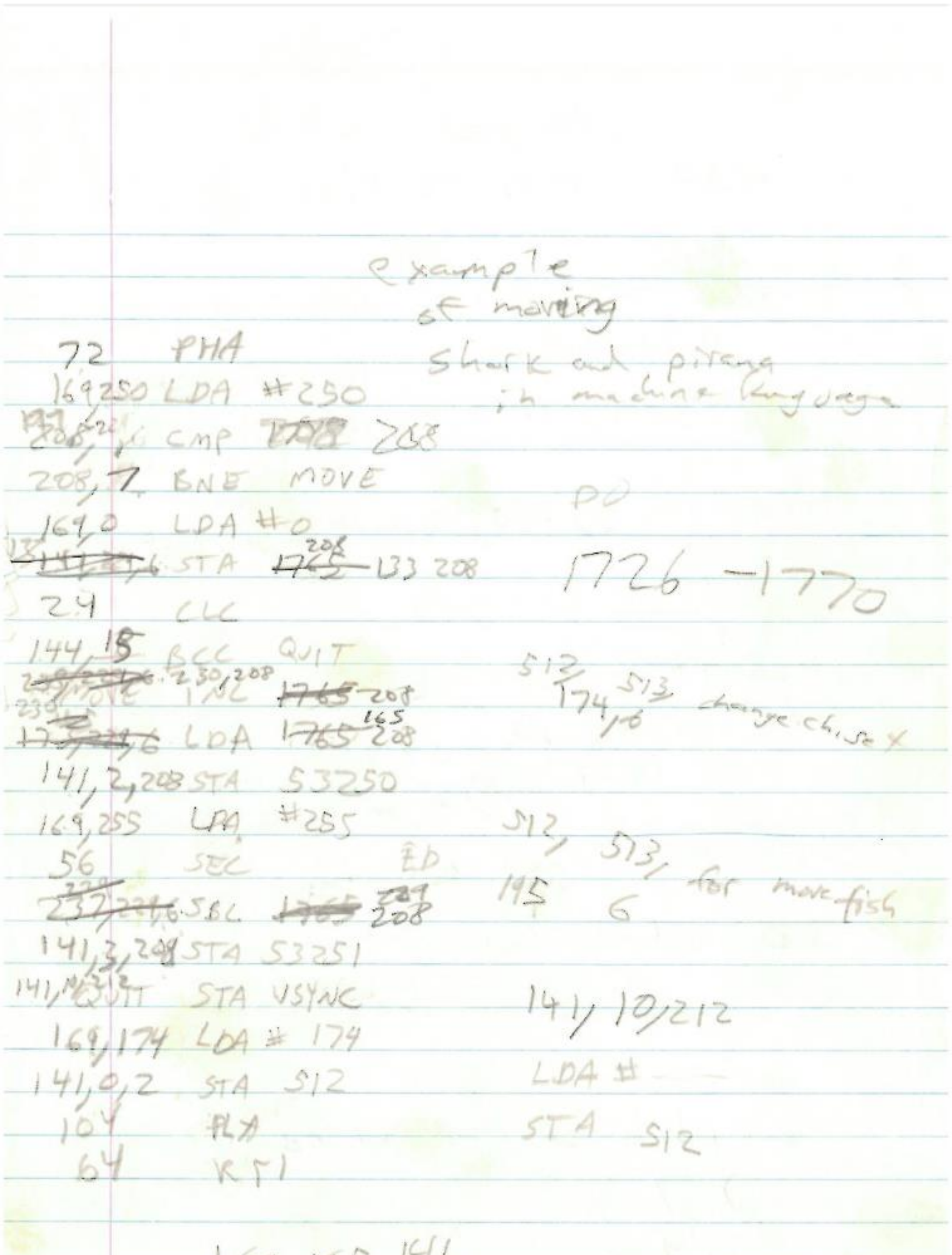
less

leftmost

2 more

14 2|2

```
QUIT JMP 58466          76, 98, 228
                        169 0 141 14 212
LEFT LDA #200           169, 186
     CMP 1650           205, 114, 6
     BNE MOVELEFT       208,   8.
     LDA #199           169, 185
     STA 1650           141, 114, 6
     CLC                24
     BCC QUIT           144, 238
MOVELEFT INC 1650       238, 114, 6
     LDA 1650           173, 114, 6
     STA 53248          141, 0, 208      53    185
     CLC                24
     BCC QUIT           144, 226
RIGHT LDA #50           169, 52
     CMP 1650           205, 114, 6
     BNE MOVERIGHT      208,   8
     LDA #51            169, 53
     STA 1650           141, 114, 6
     CLC                24
     BCC QUIT           144, 211
MOVERIGHT DEC 1650      206, 114, 6
     LDA 1650           173, 114, 6
     STA 53248          141, 0, 208
     CLC                24
     BCC QUIT           144, 199
```



example of moving shark and pirana in machine language

```
72        PHA
169250    LDA  #250
208,221,6 CMP  ~~1765~~ 208
208,7     BNE  MOVE
169,0     LDA  #0
133 ~~141,221,6~~ STA ~~1765~~ 208 133 208
24        CLC
144,18    BCC  QUIT
~~238,221,6~~ 230,208 ~~MOVE~~ INC ~~1765~~ 208
~~173,221,6~~ LDA ~~1765~~ 165 208
141,2,208 STA  53250
169,255   LDA  #255
56        SEC        ED
~~237,221,6~~ 229 SBC ~~1765~~ 229 208
141,3,208 STA  53251
141,212 QUIT STA VSYNC
169,174   LDA  # 174
141,0,2   STA  512
104       PLA
64        RTI
```

PO

1726 - 1770

512 174,0 513 change ch, so x

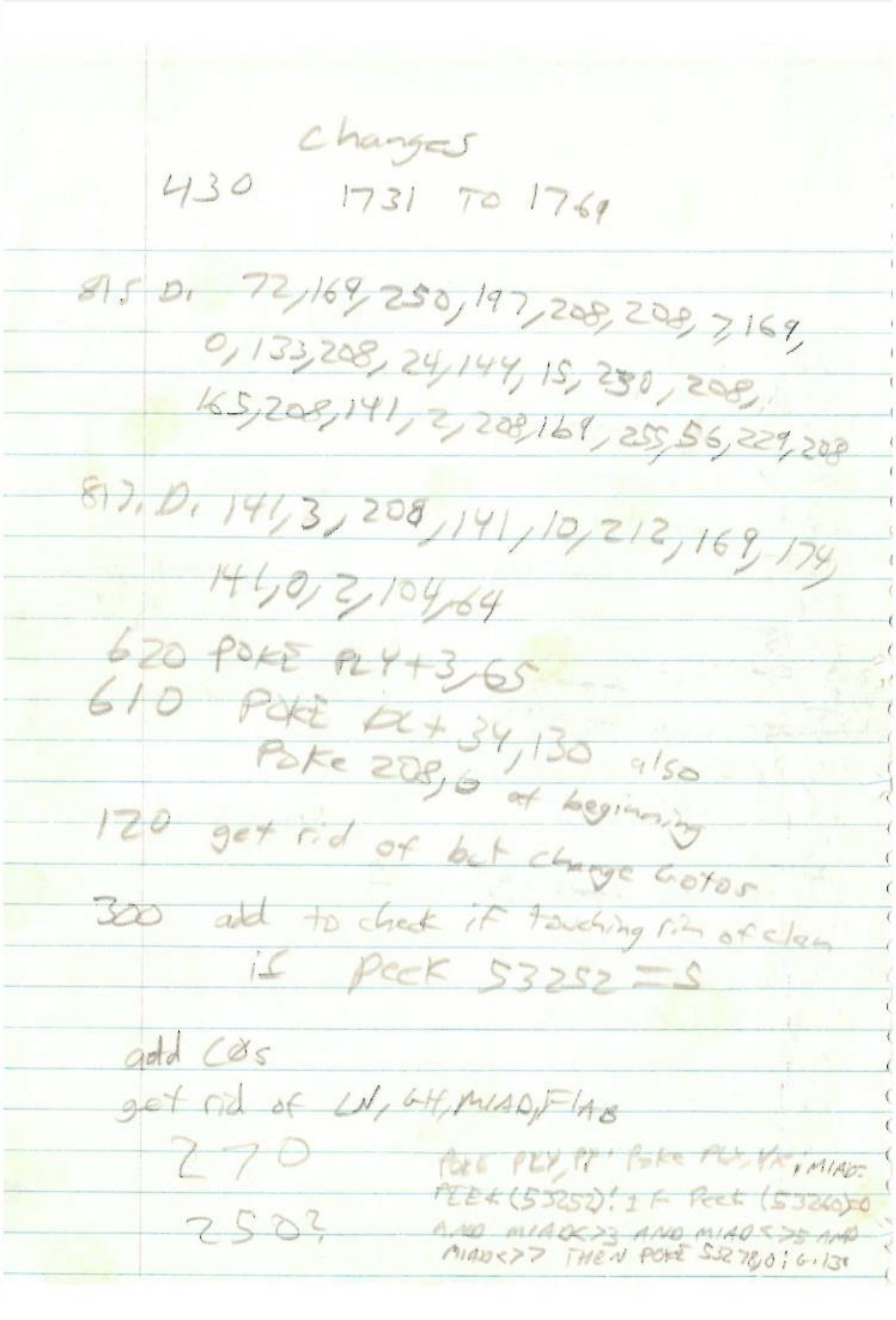
512, 513, 195 6 for move fish

141,10,212

LDA #

STA 512

169,195,141,0,2,104,64



changes

430 1731 TO 1769

815 D. 72,169,250,197,208,208,7,169,
0,133,208,24,144,15,250,208,
165,208,141,2,208,169,255,56,229,208

817. D. 141,3,208,141,10,212,169,174,
141,0,2,104,64

620 POKE PLY+3,65

610 POKE PL+34,130 also
Poke 208,0 at beginning

120 get rid of but change Gotos

300 add to check if touching rim of claw
if Peek 53252 = 5

add C&s

get rid of LN, GH, MIAD, Flag

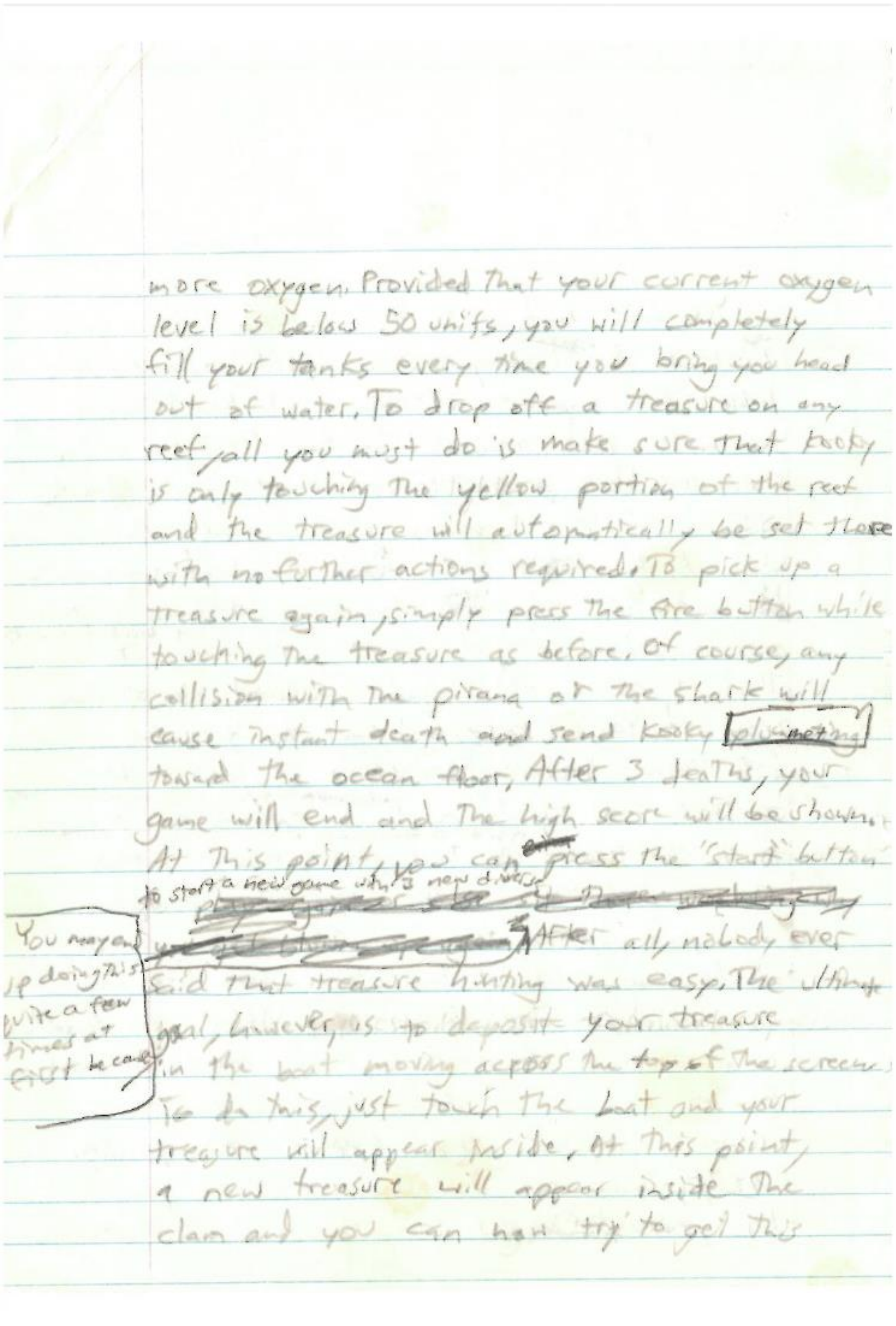
270 Poke PLY,PP: Poke PLX,VX, MIAD=
PEEK(53252): IF Peek (53260)=0

250? AND MIAD<>3 AND MIAD<>5 AND
MIAD<>7 THEN POKE 53278,0: G.130

piranha

```
            256
 256        208
 212       2048
 512      51200
 2560     532 48
 51200 12      24
 542 10     1 2
```

Yes, he's back, and he's kookier than ever, But this time,
Instead of ~~merely~~ climbing another building, Kooky has
taken the opposite extreme and has begun a
search for lost treasure. On his dives, he will
encounter black pearls, treasure chests, pistols, vases,
steering wheels, and many other treasures. But
what Kooky ~~does~~n't know as ~~when~~ he begins his expedition
is ~~is~~ that not only ~~[illegible]~~ are there a dangerous shark
and a school of man-eating pirahna, to block his course but
That a giant clam had swallowed all of the
treasure! You, as Kooky, will have to dive for
the various treasures and attempt to bring them up to the salvage boat. To move Kooky in a particular direction, you'l move the joystick in the corresponding direction. To tread water, push the joystick in any of the diagonal directions. You must time it just right
so that the clam's mouth is open when you
touch the treasure in order to have a chance to make a successful salvage. As this point, merely press
The fire button on the joystick and you
will pick up the treasure. Of course if you
touch any part of the clam when its mouth
is closed, you will lose a life. After you have
a treasure, the hard part really begins! Depending
on the weight of the treasure, you will use
up your limited oxygen supply rather quickly.
Luckily, there are 3 coral reefs for you to
lay your treasures on while surfacing for



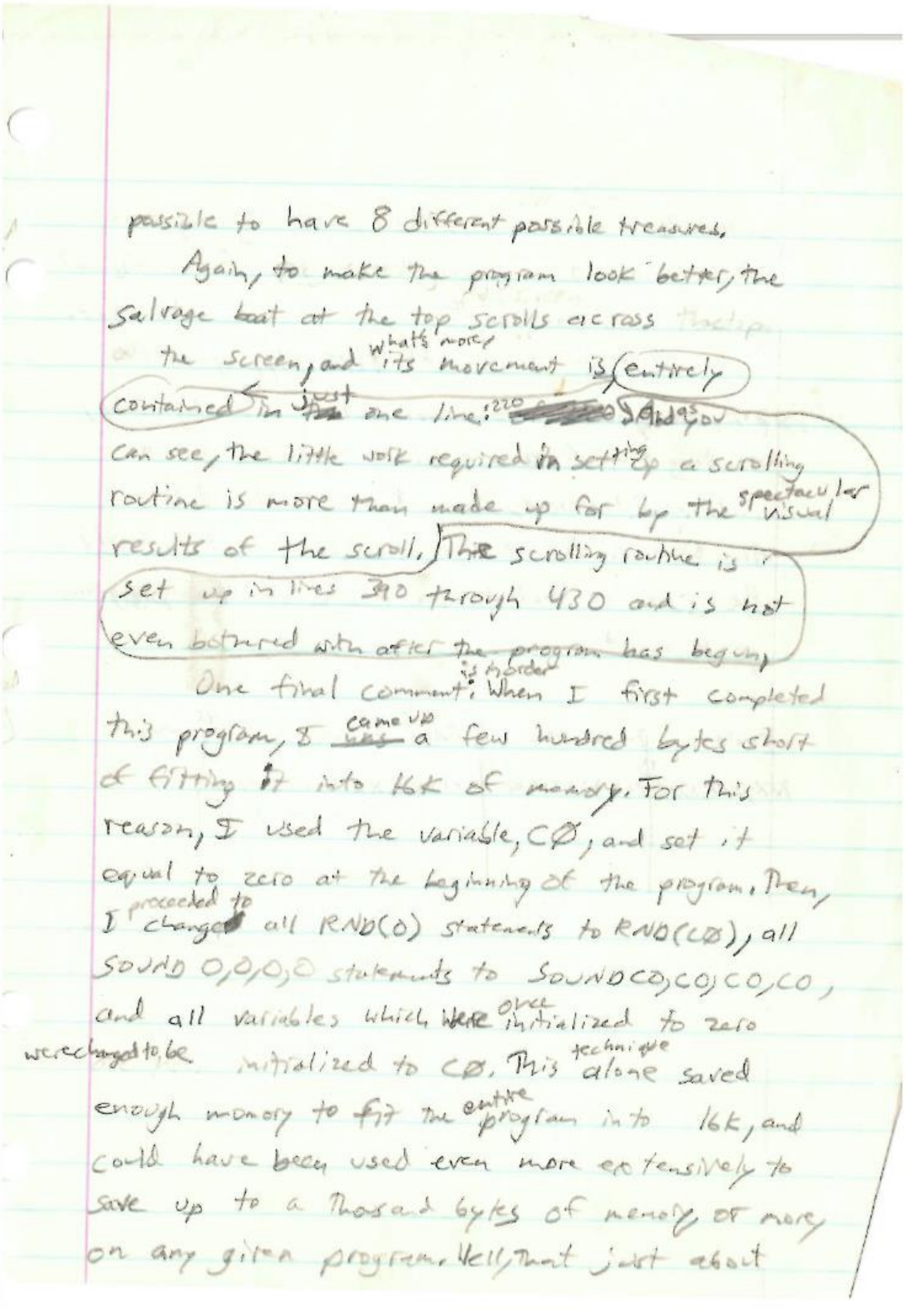
more oxygen. Provided that your current oxygen level is below 50 units, you will completely fill your tanks every time you bring you head out of water. To drop off a treasure on any reef, all you must do is make sure that Kooky is only touching the yellow portion of the reef and the treasure will automatically be set there with no further actions required. To pick up a treasure again, simply press the fire button while touching the treasure as before. Of course, any collision with the pirana or the shark will cause instant death and send Kooky plummeting toward the ocean floor. After 3 deaths, your game will end and the high score will be shown. At this point, you can ~~either~~ press the "start" button to start a new game with 3 new divers. After all, nobody ever said that treasure hunting was easy. The ultimate goal, however, is to deposit your treasure in the boat moving across the top of the screen. To do this, just touch the boat and your treasure will appear inside. At this point, a new treasure will appear inside the clam and you can now try to get this

You may end up doing this uite a few times at first becau treasure to the boat. Now that you know how to play the game, you can either type in the program and begin playing immediately, or you can read the rest of this article to learn just how Kooky Diver was written.

This program uses the machine language vertical player routine written by Tom Sak and Sid Meyer (August '82 Compute) and combines it with 2 display list interrupts in order to make an attractive, fast moving, arcade game. The first display list interrupt (figure 1) occurs between the last antic 4 line, which includes the bottom of the clam, and the line of Graphics 0 which displays the player's oxygen left, current score, and number of divers remaining. This interrupt is used to change the character set pointer at location 54281 in such a way that the upper part of the screen will continue to display the redefined characters that are used to draw the boat, coral reefs, and clam, while the lower line of graphics zero will use the standard character set located at ROM location 57344. During this interrupt, the color register at location 53272 is changed to produce a color of light green for the background of the text line, and the display list interrupt pointer is modified to point to the second display list interrupt which occurs on the next line.

(Figure 2)
This second interrupt is really the key to making the program look graphically appealing. ~~Without this interrupt~~ In order to make the school of piranas and the shark move at the same speed without the use of this interrupt, BASIC would have to be ~~[illegible]~~ used to move both of them. ~~which~~ This would both slow down the rest of the program and cause a choppy jump of 6 pixels for ~~the~~ both of the fish. As a first time machine language programmer myself, without even the aid of an assembler, I decided that I would try to code this rather simple routine in machine language, and, as you can see, I successfully managed to do so. As in the first routine ~~interrupt~~ ~~routine~~, it was necessary to modify the display list interrupt vector to point to the other ~~routine~~ interrupt, and also to store any value in location 54282 which eliminates the scan line jump that is often seen in many display list interrupt routines.

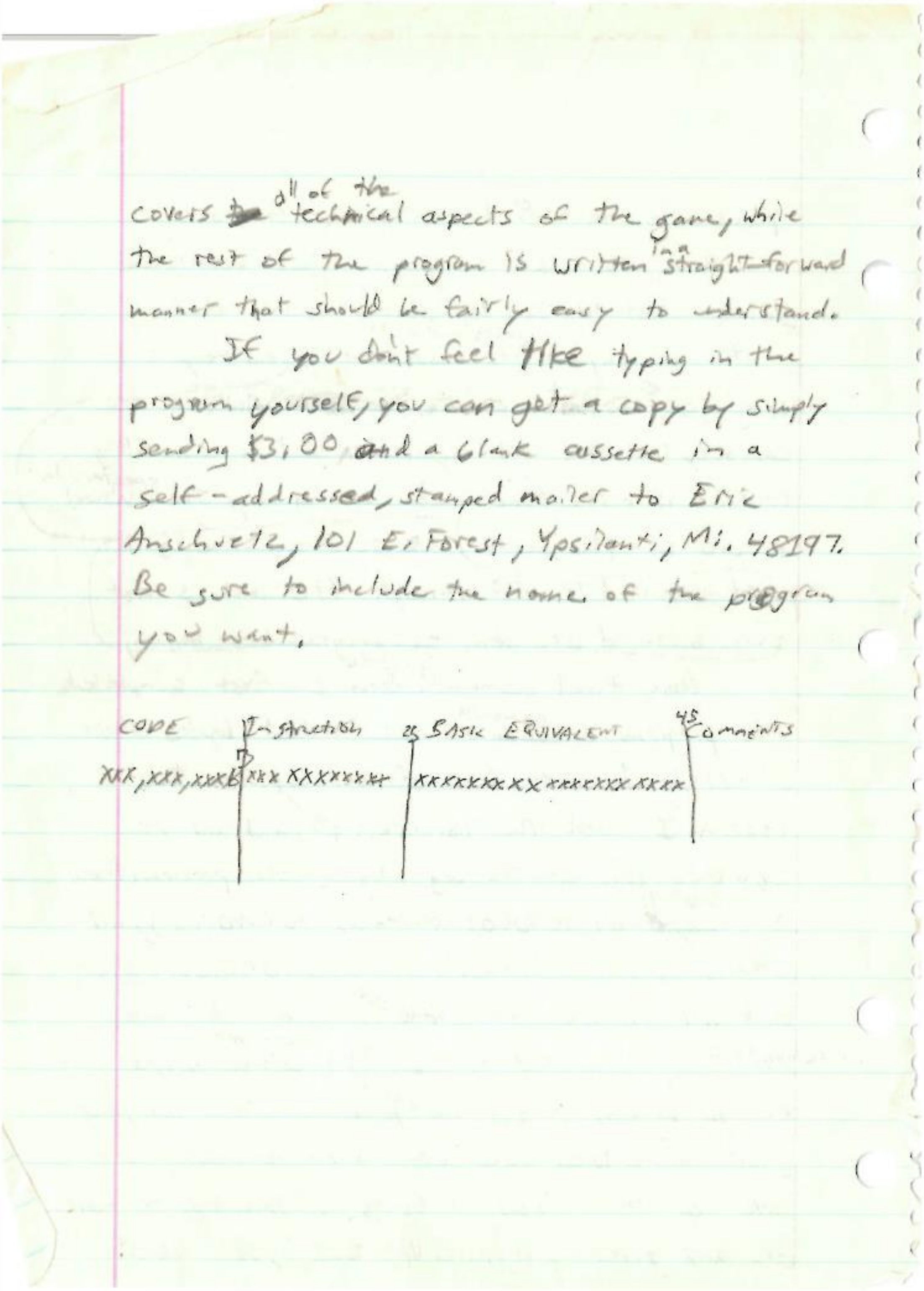
As you can see, in the program many pictures are used to draw both Kooky and the treasures that he must pick up. For this reason, at the appropriate times the values of locations PDR and PDR+1 are modified to point to the proper drawings needed. This technique was used successfully to ~~make~~ kooky animated, as well as making it



possible to have 8 different possible treasures.

Again, to make the program look better, the salvage boat at the top scrolls across the screen, and what's more, its movement is entirely contained in just ~~the~~ one line: 220 [illegible] 240. As you can see, the little work required in setting up a scrolling routine is more than made up for by the spectacular visual results of the scroll. The scrolling routine is set up in lines 390 through 430 and is not even bothered with after the program has begun.

One final comment is in order: When I first completed this program, I ~~was~~ came up a few hundred bytes short of fitting it into 16K of memory. For this reason, I used the variable, CØ, and set it equal to zero at the beginning of the program. Then, I proceeded to change all RND(0) statements to RND(CØ), all SOUND 0,0,0,0 statements to SOUND CO,CO,CO,CO, and all variables which ~~were~~ once initialized to zero were changed to be initialized to CØ. This technique alone saved enough memory to fit the entire program into 16K, and could have been used even more extensively to save up to a thousand bytes of memory or more on any given program. Well, that just about



covers ~~the~~ all of the technical aspects of the game, while the rest of the program is written in a straight-forward manner that should be fairly easy to understand.

If you dont feel like typing in the program yourself, you can get a copy by simply sending $3,00 and a blank cassette in a self-addressed, stamped mailer to Eric Anschuetz, 101 E, Forest, Ypsilanti, Mi. 48197. Be sure to include the name of the program you want,

```
CODE        |Instruction   25 BASIC EQUIVALENT        45|COMMENTS
            17
XXX,XXX,XXXB|XXX XXXXXXXX  |XXXXXXXXXX XXXXXXX XXXX  |
            |              |                         |
            |              |
```

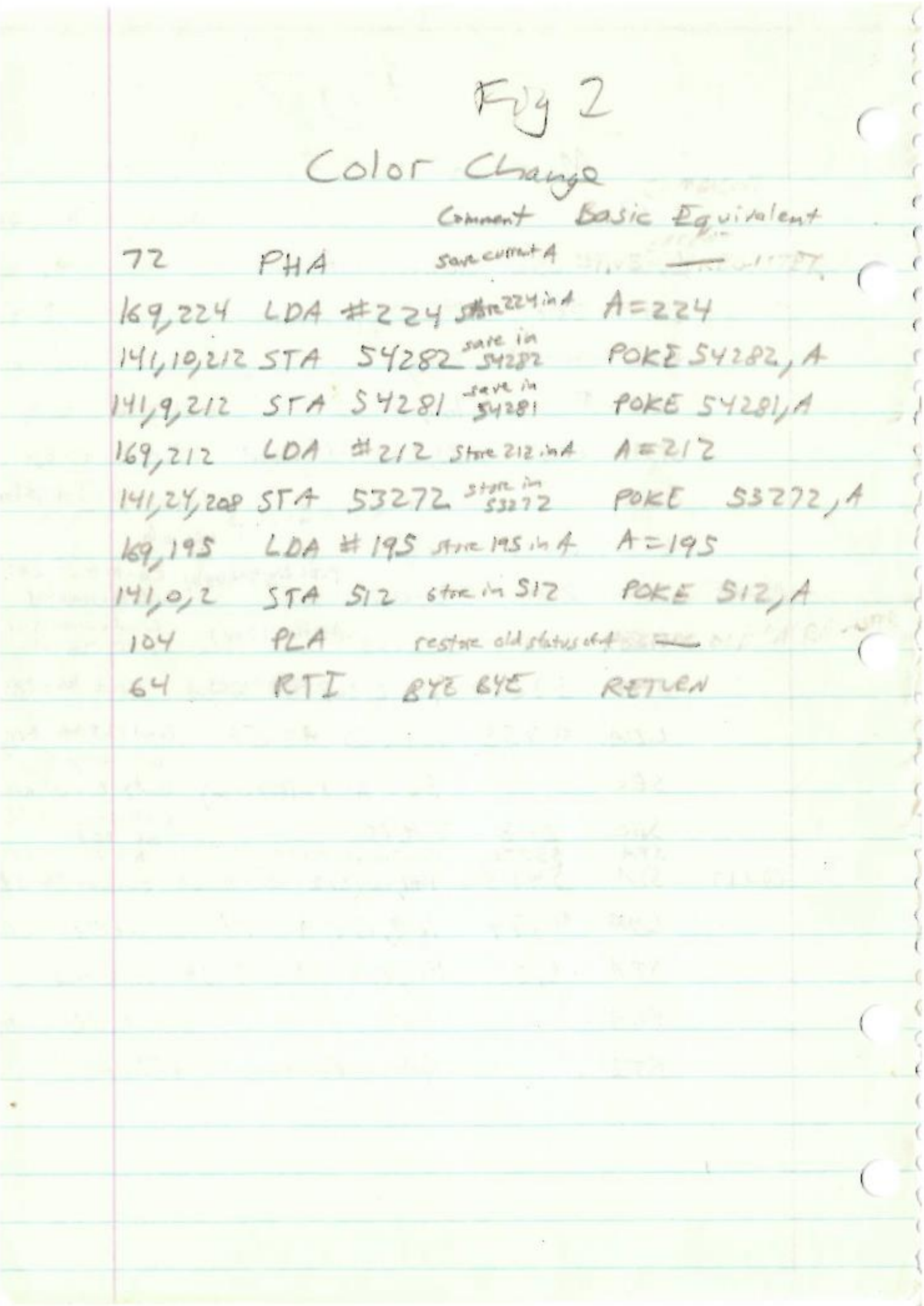


Fig 2

Color Change

		Comment	Basic Equivalent
72	PHA	save current A	—
169,224	LDA #224	store 224 in A	A=224
141,10,212	STA 54282	save in 54282	POKE 54282, A
141,9,212	STA 54281	save in 54281	POKE 54281,A
169,212	LDA #212	Store 212 in A	A=212
141,24,208	STA 53272	store in 53272	POKE 53272, A
169,195	LDA #195	store 195 in A	A=195
141,0,2	STA 512	store in 512	POKE 512, A
104	PLA	restore old status of A	—
64	RTI	BYE BYE	RETURN

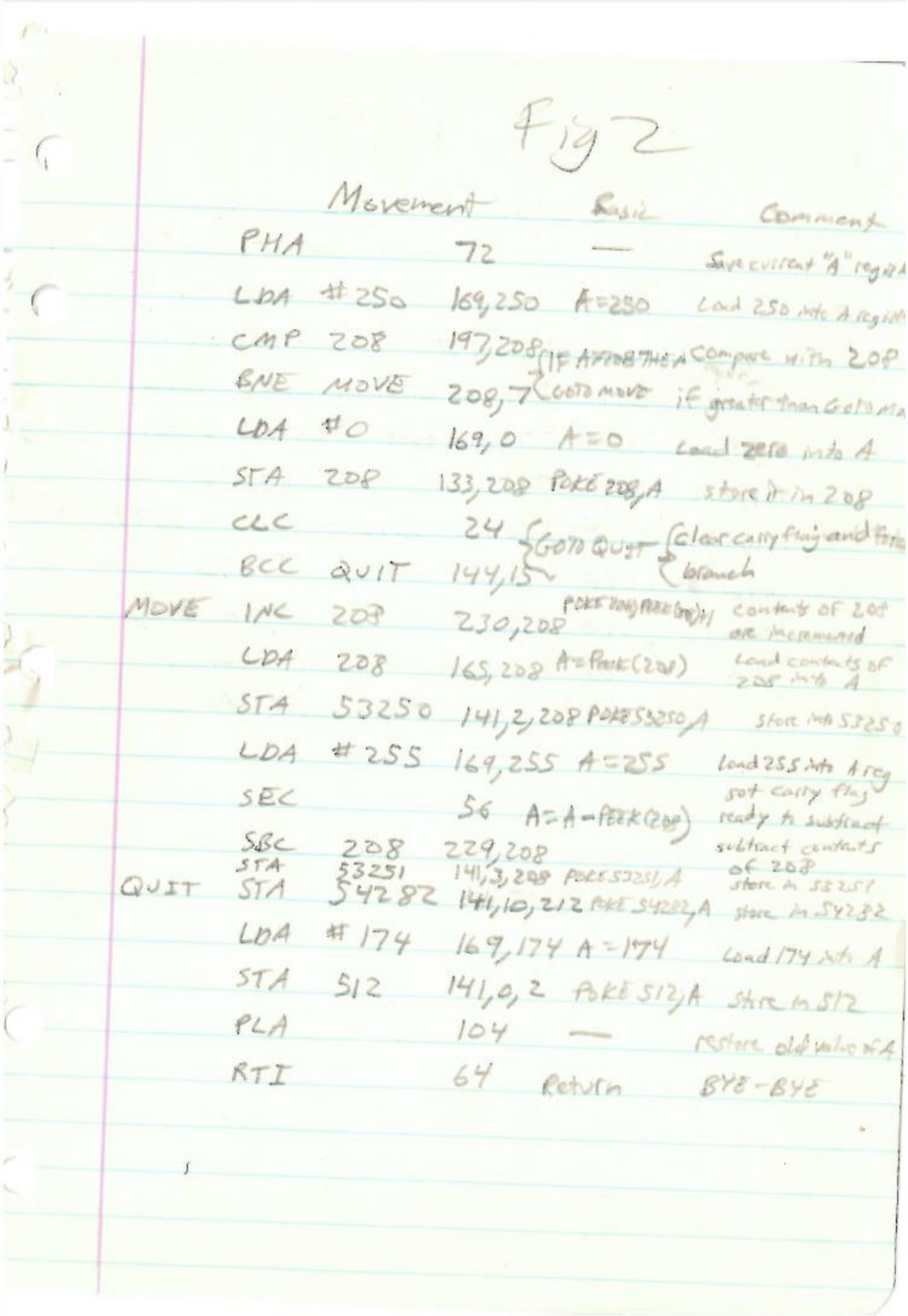
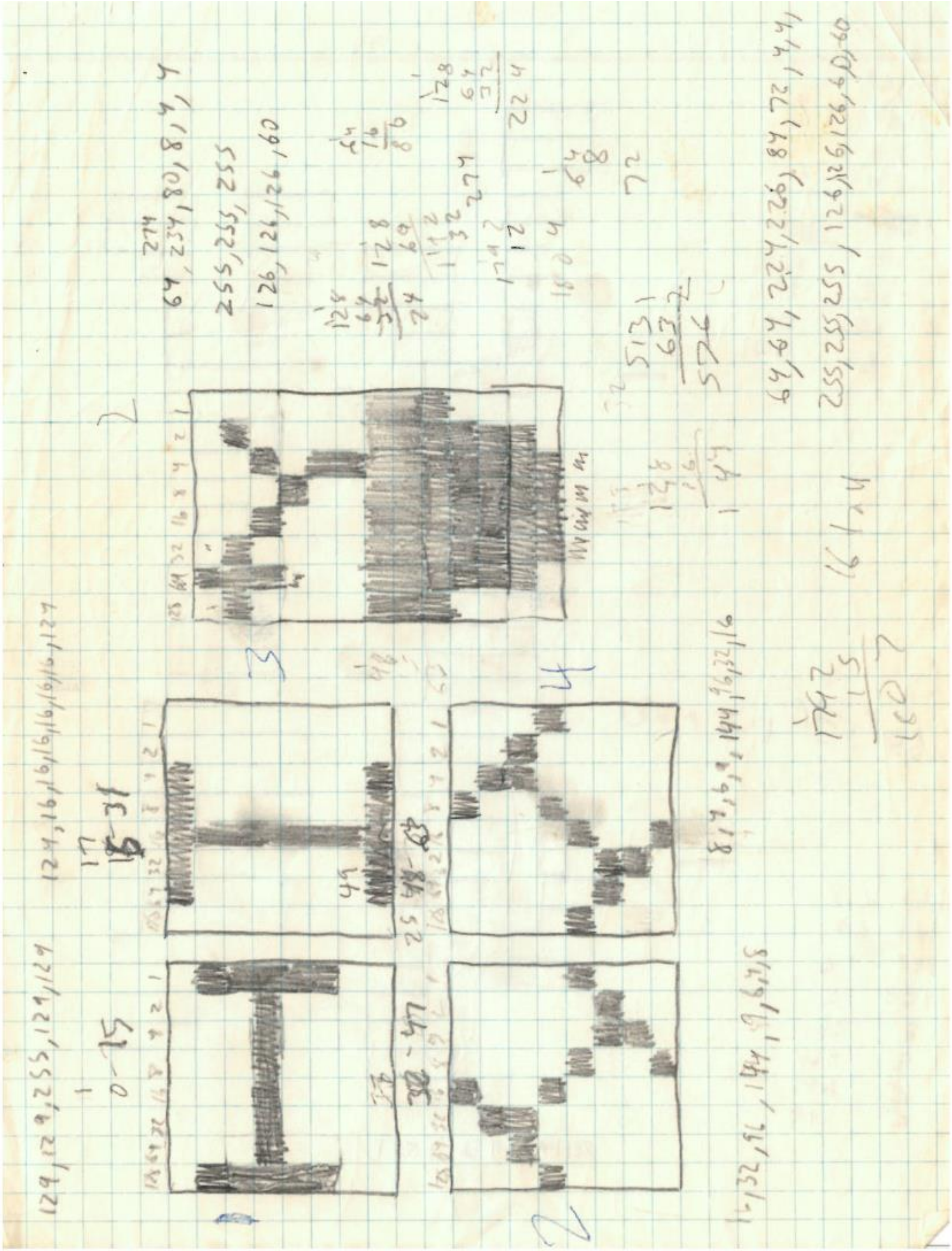


Fig 2

	Movement			Basic	Comment
	PHA		72	—	Save current "A" register
	LDA	#250	169,250	A=250	Load 250 into A register
	CMP	208	197,208	IF A>208 THEN	Compare with 208
	BNE	MOVE	208,7	GOTO MOVE	if greater than Goto Mo
	LDA	#0	169,0	A=0	Load zero into A
	STA	208	133,208	POKE 208,A	store it in 208
	CLC		24	GOTO QUIT	Clear carry flag and force
	BCC	QUIT	144,15		branch
MOVE	INC	208	230,208	POKE 208,PEEK(208)+1	contents of 208 are incremented
	LDA	208	165,208	A=PEEK(208)	Load contents of 208 into A
	STA	53250	141,2,208	POKE 53250,A	store into 53250
	LDA	#255	169,255	A=255	load 255 into A reg
	SEC		56	A=A-PEEK(208)	set carry flag ready to subtract
	SBC	208	229,208		subtract contents of 208
	STA	53251	141,3,208	POKE 53251,A	store in 53251
QUIT	STA	54282	141,10,212	POKE 54282,A	store in 54282
	LDA	#174	169,174	A=174	Load 174 into A
	STA	512	141,0,2	POKE 512,A	Store in 512
	PLA		104	—	restore old value of A
	RTI		64	Return	BYE-BYE

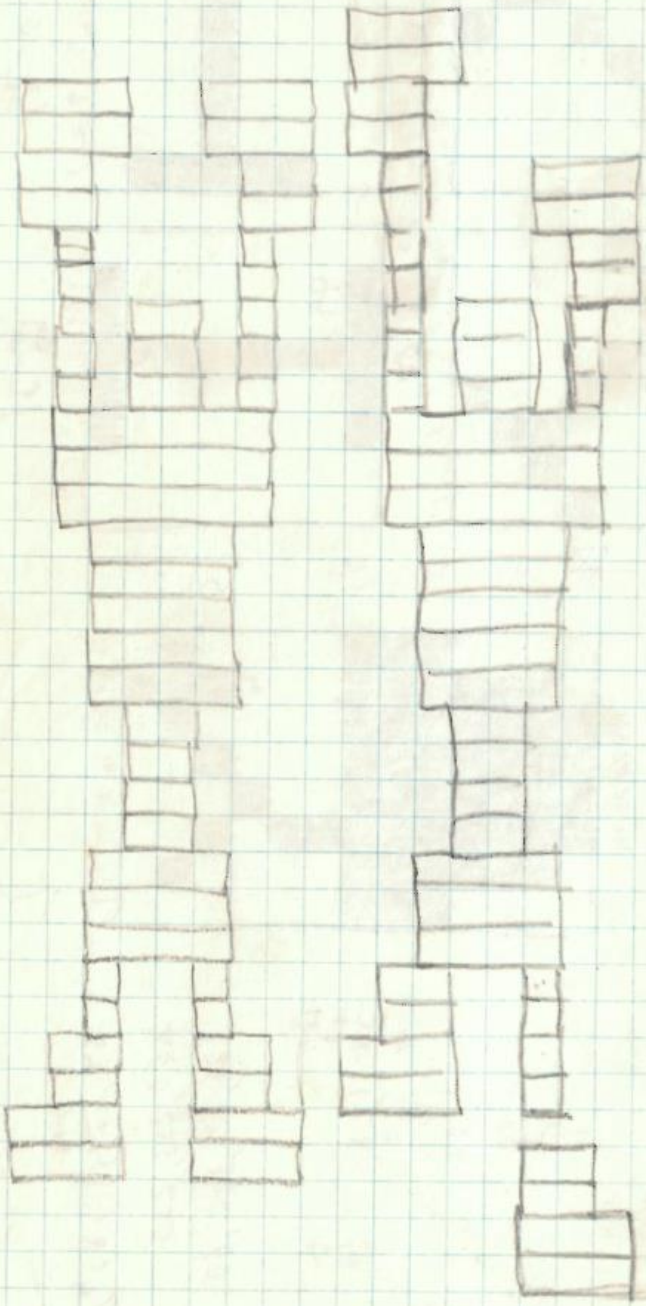
KOOKY KLIMBER



64

1 2 4 8 16 32 64 128

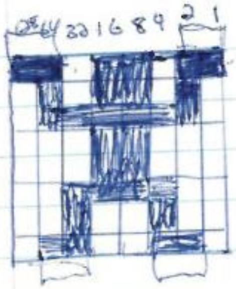
224	7
224	7
192	3
192	3
71	226
71	226
67	144
67	19



Draw = 1, 35, 69

100	38
100	38
228	39
228	39
~~4~~	
6	96
6	96
7	224
7	224

224
231

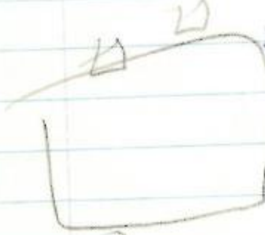


231

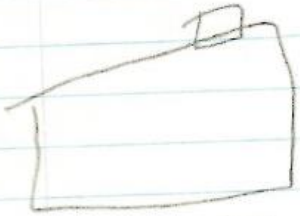
231 195
~~219~~ 195
~~219~~
66
66
90
90 } head
90
126
126 126
60
60 60, 60, 60
60 60,
24
24
24
~~24~~ 60
60
60
36
36
36
102 102
231

231

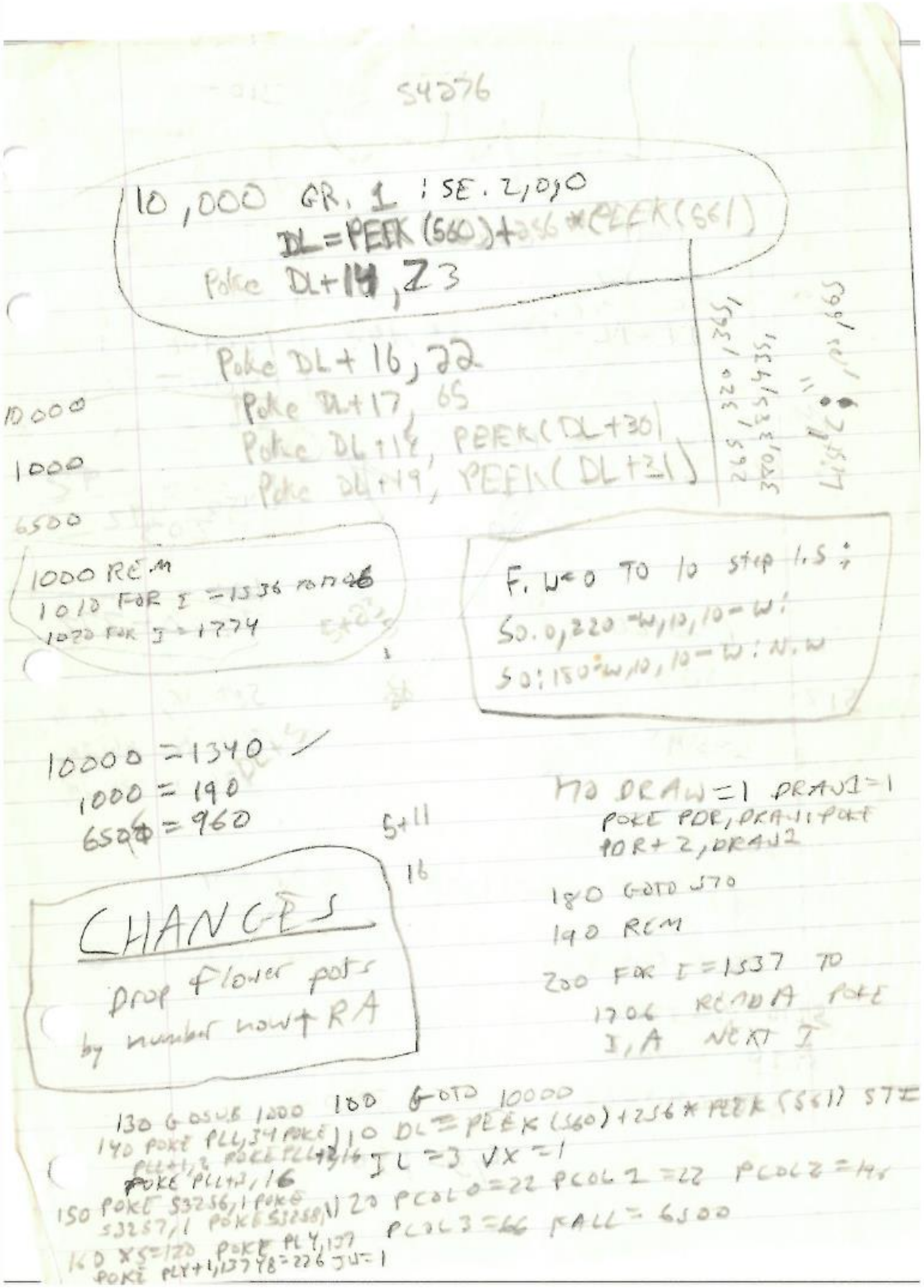
219
-24
195



224
224
192
192
64
64
61
67
61
91
90
90



0h
8
ZΣ



54276

```
10,000 GR. 1 : SE. 2,0,0
        DL=PEEK (560)+256*PEEK(561)
    Poke DL+14, 23

    Poke DL+16, 22
     Poke DL+17, 65
     Poke DL+18, PEEK(DL+30)
      Poke DL+19, PEEK(DL+31)
```

265, 320, 365,
370,335/345/355,
List 1537 : " ,560,565

10000
1000
6500

```
1000 REM
1010 FOR I =1536 TO 1746
1020 FOR J =1774
```

```
F. W=0 TO 10 step 1.5 :
SO. 0,220-W,10,10-W :
SO. 150-W,10,10-W : N. W
```

```
10000 =1340
 1000 = 190
 6500 = 960
```

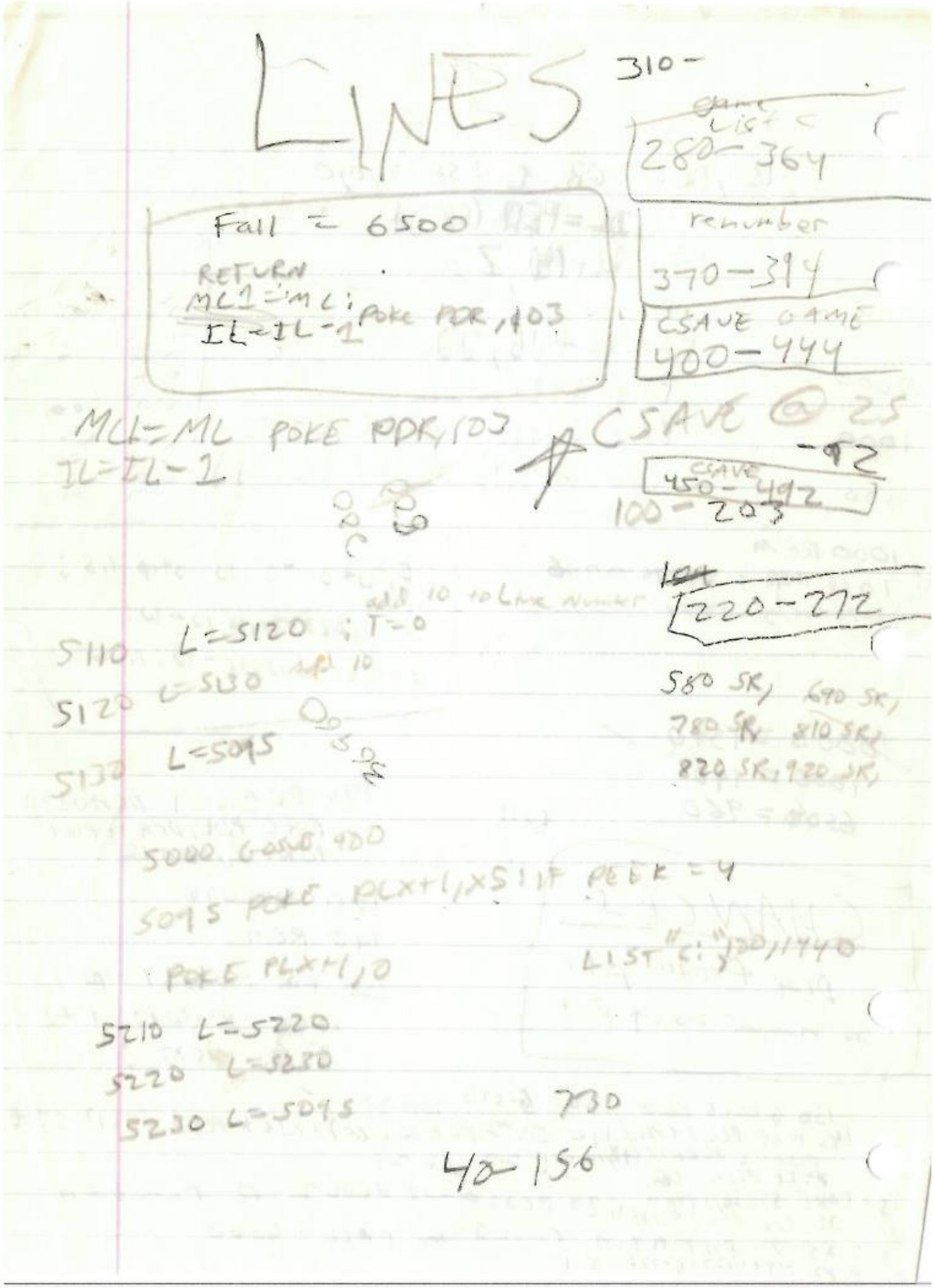
5+11
16

CHANGES

Drop flower pots by number now↑ RA

```
170 DRAW=1 DRAW2=1
    POKE PDR,DRAW1 POKE
    PDR+2,DRAW2
180 GOTO 570
190 REM
200 FOR I =1537 TO
    1706 READ A POKE
    I,A NEXT I
```

```
130 GOSUB 1000   100 GOTO 10000
140 POKE PLL,34 POKE  110 DL=PEEK (560)+256*PEEK (561) ST=
   PLL+1,2 POKE PLL+3,16  IL=3 VX=1
   POKE PLL+2,16
150 POKE 53256,1 POKE  120 PCOL0=22 PCOL1=22 PCOL2=148
   53257,1 POKE 53258,1   PCOL3=66 FALL=6500
160 XS=120 POKE PLY,137
   POKE PLY+1,137 YS=226 JU=1
```



LINES 310-

~~game~~
List <
280-364

renumber
370-394

CSAVE GAME
400-444

Fall = 6500

RETURN
ML1=ML: Poke PDR,103
IL=IL-1

ML1=ML POKE PDR,103
IL=IL-1

CSAVE @ 25
-92

CSAVE
450-492

100-203

~~104~~
220-272

add 10 to Line Number

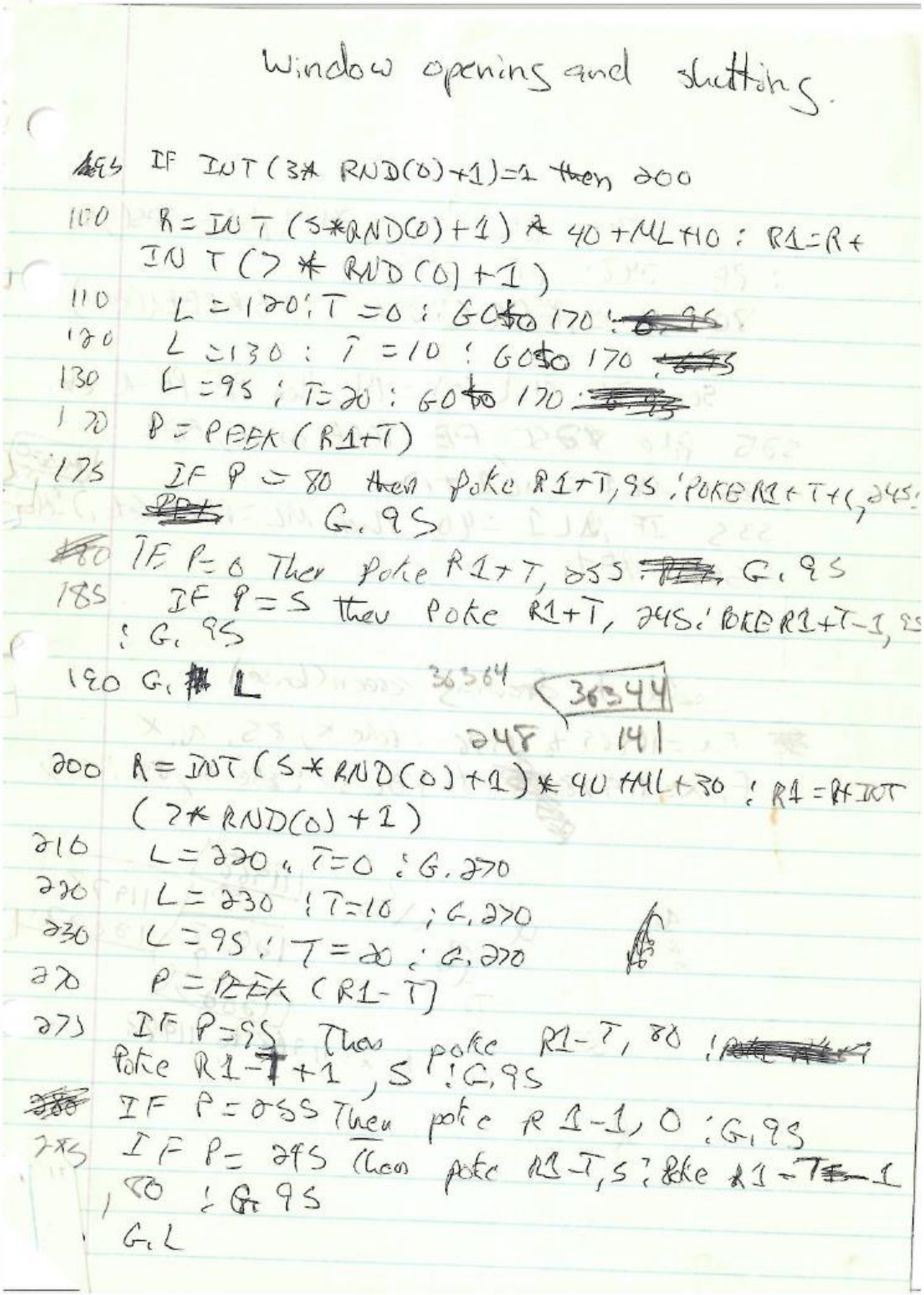
```
5110 L=5120 : T=0
5120 L=5130 add 10
5130 L=5095

5000 GOSUB 900
5095 POKE PLX+1,XS1 IF PEEK = 4
     POKE PLX+1,0

5210 L=5220
5220 L=5230
5230 L=5095
```

580 SK, 690 SK,
780 SK, 810 SK,
820 SK, 920 SK,

LIST "C:",130,1440

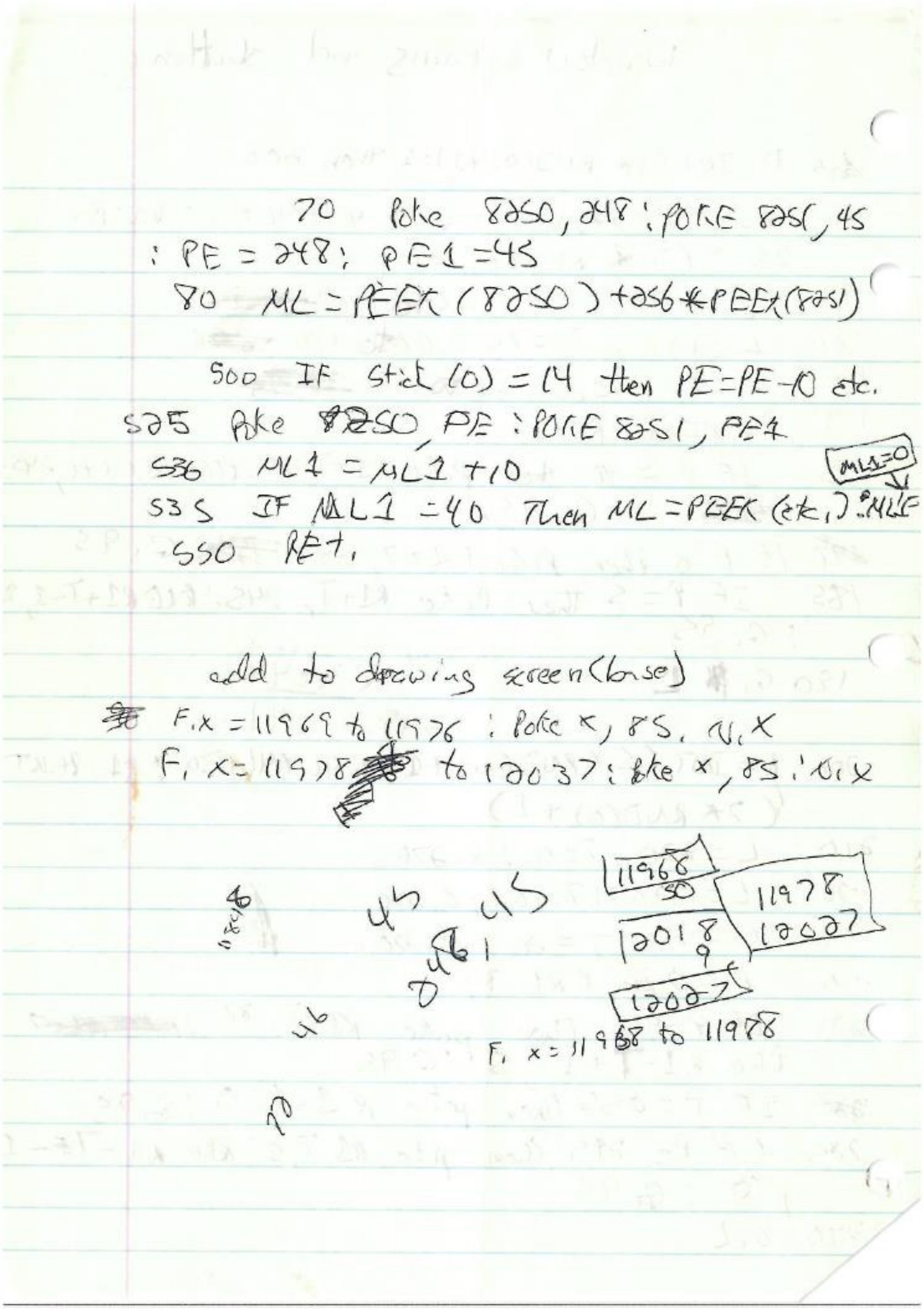
730

40-156



Window opening and shutting.

```
95 IF INT(3* RND(0)+1)=1 then 200
100 R=INT(5*RND(0)+1) * 40+ML+10 : R1=R+
    INT(7 * RND(0)+1)
110  L=120: T=0 : GOTO 170 : ~~G 95~~
120  L=130 : T=10 : GOTO 170 ~~G 95~~
130  L=95 ; T=20 : GOTO 170 : ~~G 95~~
170 P=PEEK(R1+T)
175  IF P=80 then Poke R1+T,95 : POKE R1+T+1, 245:
    ~~PEE~~      G.95
180 IF P=0 Then Poke R1+T, 255 : ~~PEE~~ G.95
185   IF P=5 then Poke R1+T, 245 : POKE R1+T-1, 25
    : G.95
190 G. L
                    36364   36344
                     248     141
200 R=INT(5*RND(0)+1)*40+ML+30 : R1=R+INT
    (7*RND(0)+1)
210   L=220 : T=0 : G.270
220   L=230 : T=10 ; G.270
230   L=95 : T=20 : G.270
270   P=PEEK(R1-T)
275  IF P=95 Then poke R1-T, 80 : [illegible]
    Poke R1-T+1, 5 : G.95
280  IF P=255 Then poke R1-1, 0 : G.95
285  IF P=245 Then poke R1-T,5 : Poke R1-T-1
    , 80 : G.95
    G.L
```



```
      70  Poke 8250,248: POKE 8251,45
: PE = 248: PE1 =45
 80  ML = PEEK (8250) +256*PEEK(8251)

    500 IF Stick (0) = 14 then PE=PE-10 etc.
525 Poke 8250, PE : POKE 8251, PE1
 536  ML1 = ML1 +10
 535  IF ML1 =40  Then ML = PEEK (etc.) : ML1=0
 550  RET.
```

add to drawing screen(base)

```
F.x = 11969 to 11976 : Poke x, 85. N.X
F. x= 11978 to 12037: Poke x, 85 : N.X
```

45 45

246

46

70

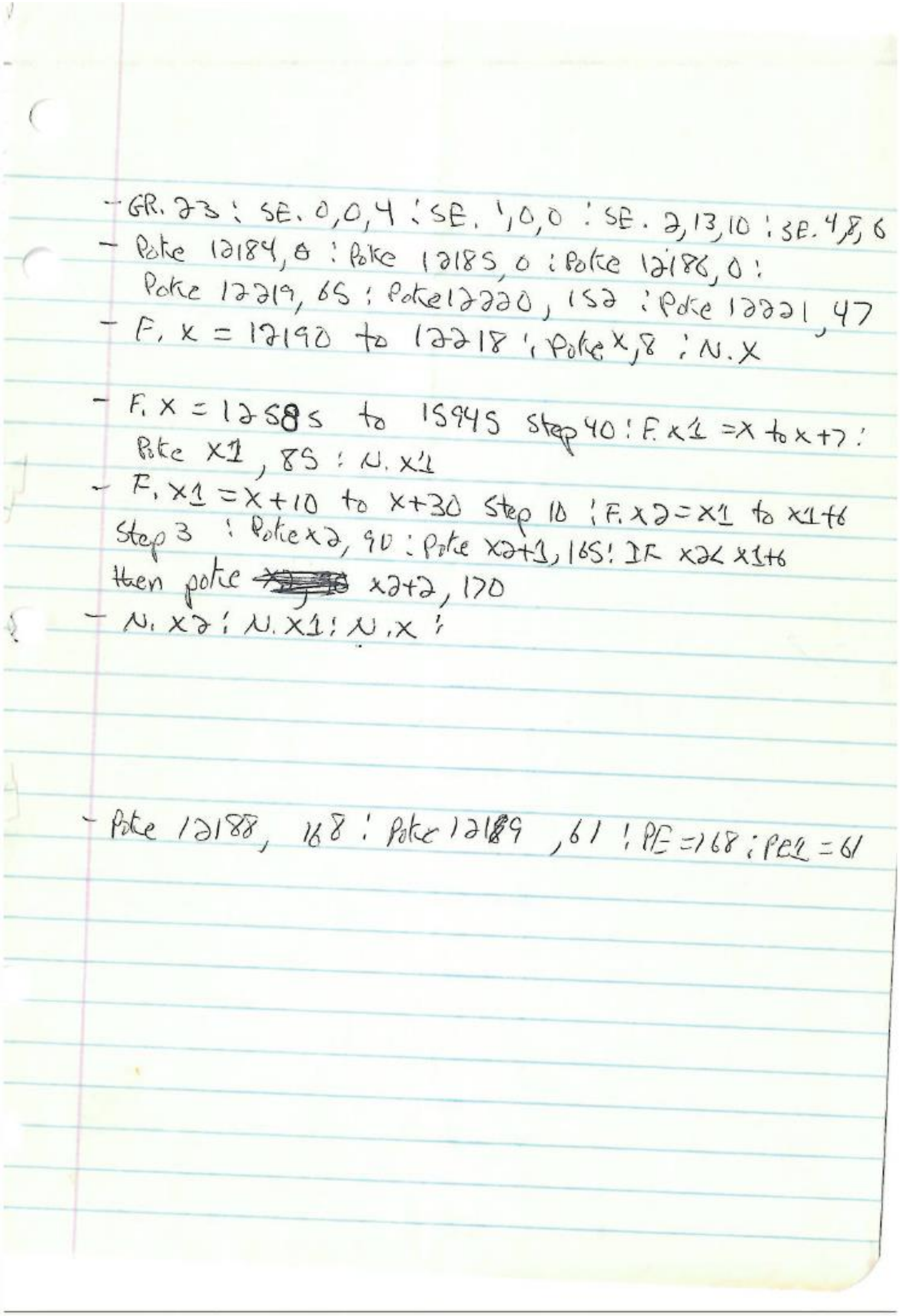
11968
50

11978
12037

12018
9

12027

F. x = 11988 to 11978



```
- GR.23 : SE.0,0,4 : SE.1,0,0 : SE.2,13,10 : SE.4,8,6
- Poke 12184,0 : Poke 12185,0 : Poke 12186,0 :
  Poke 12219,65 : Poke12220,152 : Poke 12221,47
- F. X = 12190 to 12218 : Poke X,8 : N.X

- F. X = 12585 to 15945 Step 40 : F. X1 = X to X+7 :
  Poke X1,85 : N. X1
- F. X1 = X+10 to X+30 Step 10 : F. X2 = X1 to X1+6
Step 3 : Poke X2,90 : Poke X2+1,165 : IF X2< X1+6
then poke X2+2,170
- N. X2 : N. X1 : N. X :




- Poke 12188, 168 : Poke 12189 ,61 : PE=168 : PE1=61
```

(DL= 36760
36862   548   36314

ML = Peek (DL+4)+256 * PEEK(DL+5)

36314

DL -446   DL+3600

Poke DL,0 : Poke DL+1,0 : Poke DL+2,0 :
F. X= DL+6 to DL+34 : Poke X,8 : N.X
Poke DL+35, 65 : Poke DL+36, PEEK(DL+102)
: Poke DL+37, PEEK(DL+103)

F. X=DL+401 to DL+3761

40300

$\boxed{128}$ $\boxed{152}$

40320

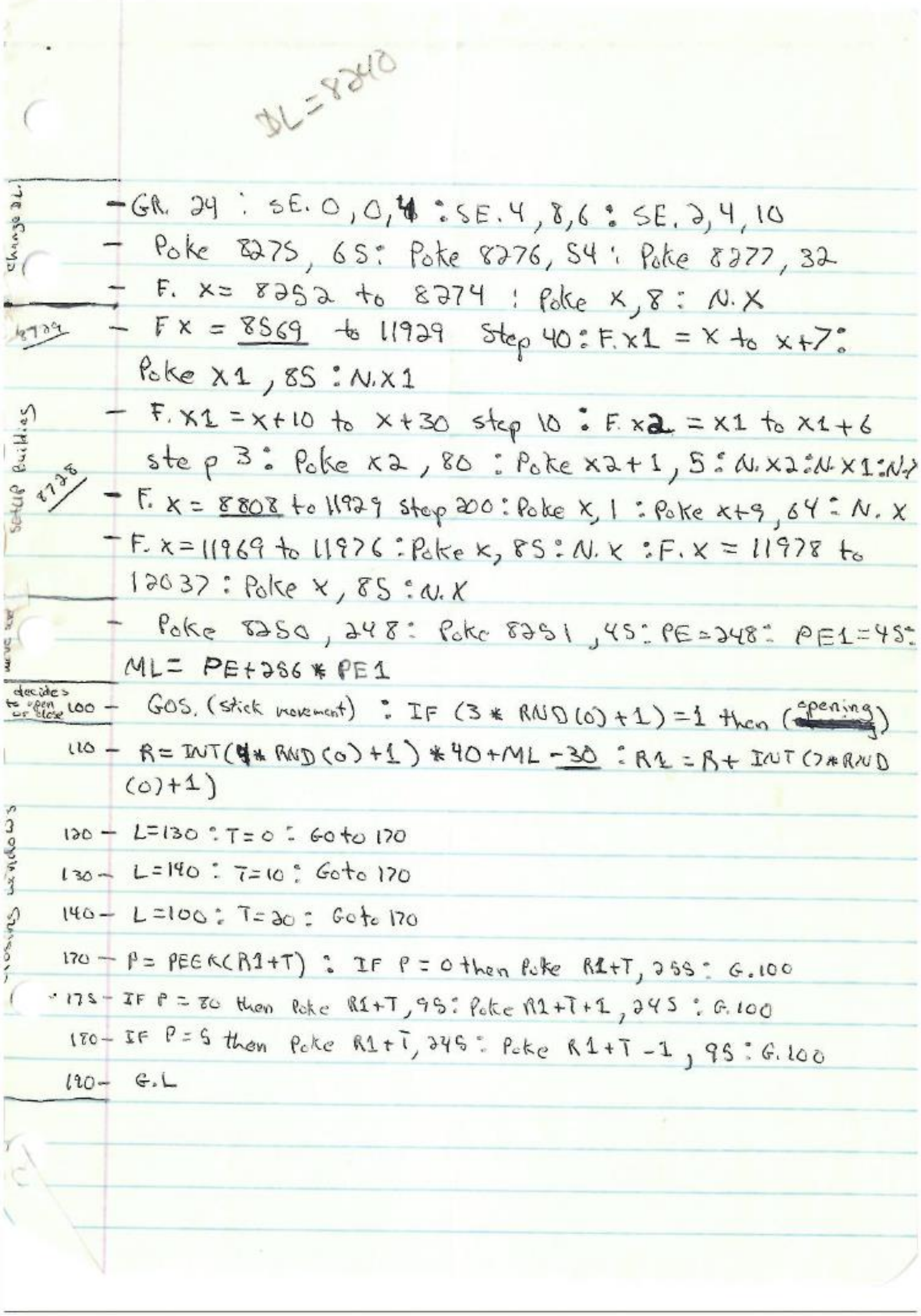
218   141

3560

Poke DL+4, DL+3560 - X *256   INT(DL+3560/256)
Poke DL+5 ,X

X=INT(DL+3560/256)   Poke DL+5, INT(DL+3560/256

Poke DL+4, DL+3560 - (INT *256



DL=8240

change DL

```
-GR. 24 : SE.0,0,4 : SE.4,8,6 : SE.2,4,10
 - Poke 8275, 65: Poke 8276, 54 : Poke 8277, 32
 - F. X= 8252 to 8274 : Poke X,8 : N.X
```

8729

setup buildings

8728

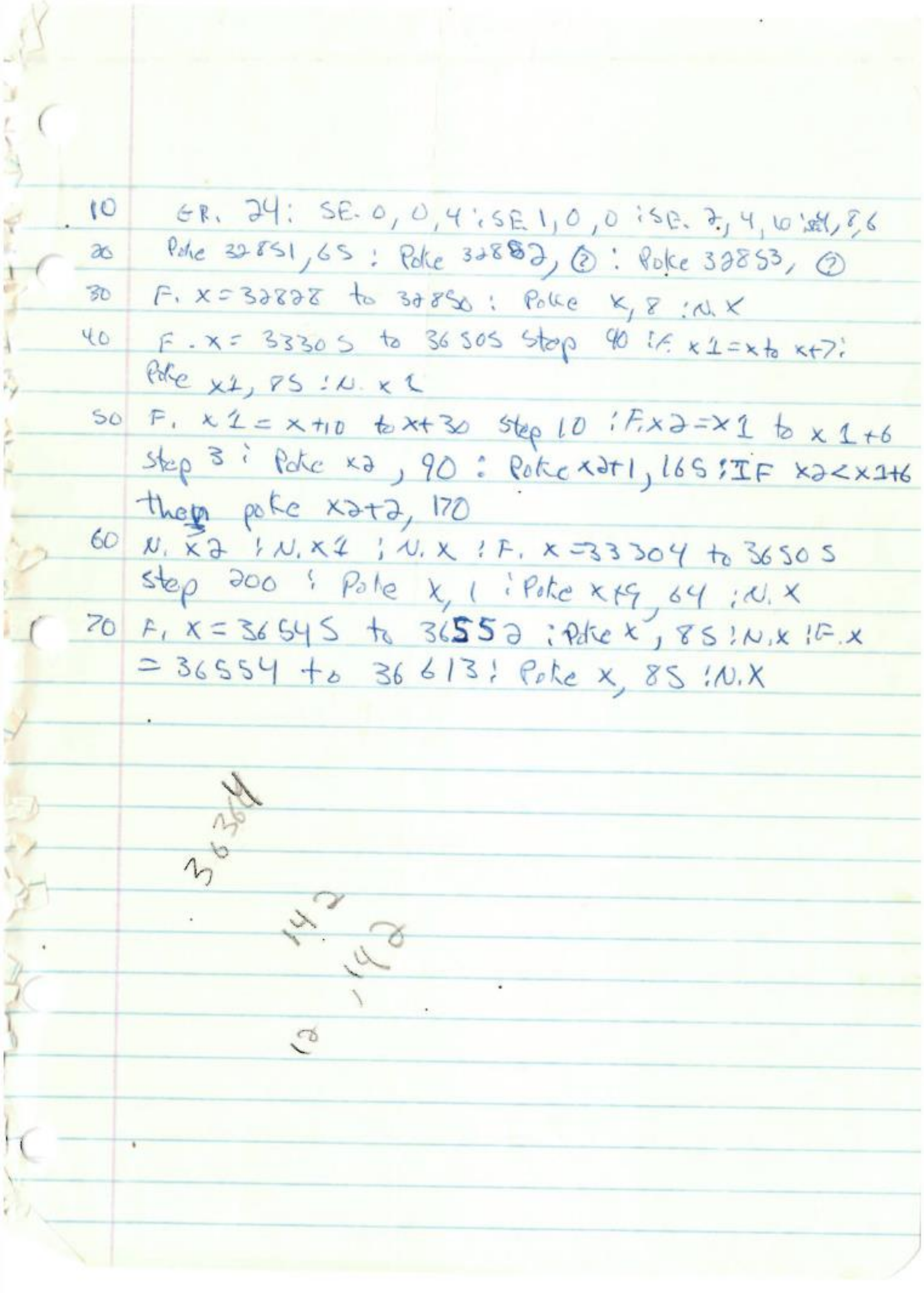
```
 - FX = 8569 to 11929 Step 40 : F.X1 = X to X+7:
   Poke X1, 85 : N.X1
 - F.X1 = X+10 to X+30 step 10 : F.X2 = X1 to X1+6
   step 3: Poke X2, 80 : Poke X2+1, 5 : N.X2:N.X1:N.X
 - F. X = 8808 to 11929 step 200 : Poke X, 1 : Poke X+9, 64 : N. X
 - F. X=11969 to 11976 : Poke X, 85 : N.X : F.X = 11978 to
   12037 : Poke X, 85 : N.X
 - Poke 8250, 248: Poke 8251, 45: PE=248: PE1=45:
   ML= PE+256 * PE1
```

decides to open or close

```
100 - GOS. (stick movement) : IF (3 * RND(0)+1)=1 then (opening)
110 - R= INT(4* RND(0)+1) * 40+ML-30 : R1 = R+ INT(7*RND
      (0)+1)
```

closing windows

```
120 - L=130 : T=0 : Goto 170
130 - L=140 : T=10 : Goto 170
140 - L=100 : T=20 : Goto 170
170 - P= PEEK(R1+T) : IF P = 0 then Poke R1+T, 255 : G.100
175 - IF P = 80 then Poke R1+T, 95: Poke R1+T+1, 245 : G.100
180 - IF P = 5 then Poke R1+T, 245 : Poke R1+T -1, 95 : G.100
190 - G.L
```



```
10  GR. 24: SE.0,0,4:SE.1,0,0:SE.2,4,10:[illegible],8,6
20  Poke 32851,65 : Poke 32852, (?) : Poke 32853, (?)
30  F. X=32828 to 32850 : Poke X,8 :N.X
40  F.X= 33305 to 36505 step 40 :F. X1=X to X+7:
    Poke X1, 85 :N. X1
50  F. X1 = X+10 to X+30 step 10 :F.X2=X1 to X1+6
    step 3 : Poke X2 , 90 : Poke X2+1, 165 :IF X2<X1+6
    then poke X2+2, 170
60  N. X2 :N.X1 :N.X :F. X=33304 to 36505
    step 200 : Poke X,1 :Poke X+9,64 :N.X
70  F. X=36545 to 36552 :Poke X,85 :N.X :F.X
    =36554 to 36613: Poke X, 85 :N.X
```

36364

142

142

12

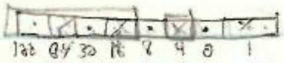
```
                                                        change
10 GR.24 : SE.00,4 : SE.4,8,6 : SE.2,4,10
20 Poke 32851,65 : Poke 32852,54 : Poke 32853,32
30 F.x=32828 to 32850 : Poke x,8 :N.X

40 F.x=33305 to 36505 Step 40 : ~~F.x1 to x1+6 Step 3:~~
   ~~Poke x2,80 : Poke x2+1,5 : N.x2~~ : F.x1=x to x+7 :
   Poke x1,85 : N.x1
50 F.x1=x+10 to x+30 Step 10 : F.x2=x1 to x1+6 Step 3 :
   Poke x2,80 : Poke x2+1,5 : N.x2 : N.x1 : N.X
60 F.X=33304 to 36505 step 200 : Poke x,1 : Poke x+9,64 : N.x
70 F.x=36545 to 36592 : Poke x,85 : N.x : F.X=36554 to
   36613 : Poke x,85 : N.X
```

64 80
16

90

5

2
128
32
8
2
170



15

100   170, 255, 85

F. ML1 to DL+3761

F x=ML1 to DL+3761
P=Peek(x)

IF P=255 then
poke x,85
IF P=95 then
poke x,90
IF P=245 then
poke x,165
NEXT X

112 192
32 48
224 144

48 — 192

32
|
224

X8, Y8 X9, Y9

6140 – 6390

~~6140 IF Y9 > 224 or Y9 > 224 THEN E=INT(RND(0)*2+1)~~
~~6145 IF E=1 THEN~~
~~6150 X9=INT(RND(0)*144+48):~~

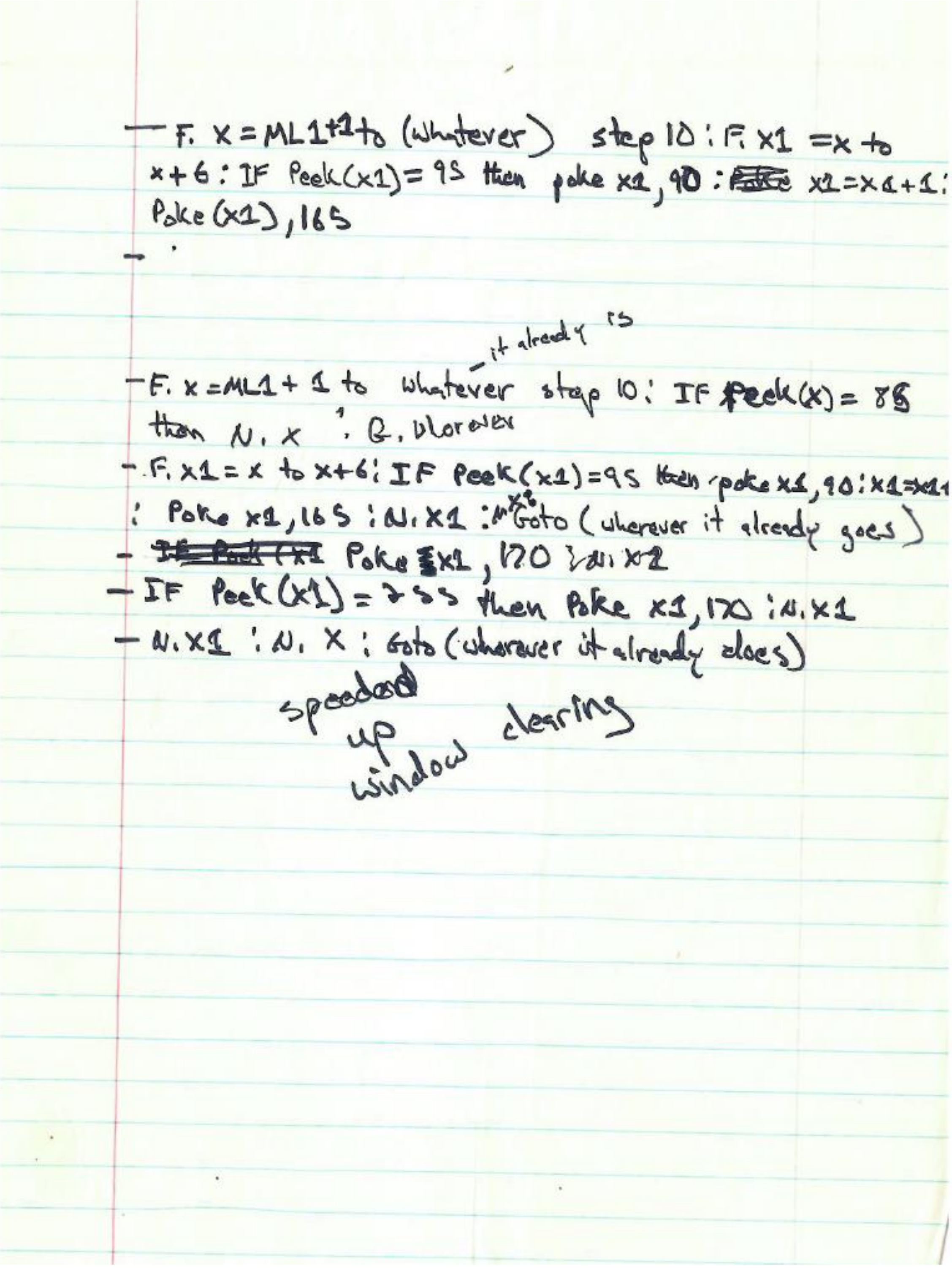
10000 ~~E~~EA=INT(RND(0)*2+1)
10010 IF EA=1 THEN X8=INT(RND(0)*144+48):Y8=10:RETURN
1020 X9=INT(RND(0)*144+48):Y9 10:~~[illegible]~~:DRAW=1:RETUR

~~6140 IF Y8 >225 or Y9 >225 THEN GOSUB 10000~~
6145 IF EA=2 THEN Y8=Y9+3: ~~IF Y8 >225 THEN GOSUB 10000~~
~~6150 Y9=Y9+3~~
IF EA=1 THEN
6180 POKE PDR+2,DRAW: Y9=Y9+3
~~[illegible]~~ DRAW=DRAW+16: ~~IF Y9>225 THEN GOSUB 10000~~
IF DRAW>50 THEN DRAW=1

49
16
‾‾
6 5

X8=0 X9=0
Y8=0 Y9=0
Poke PLX+2,0
Y+2,0
Y+3,0
X+3,0

6140 IF Y8>225 or Y9>225 THEN GOSUB 6170
6145 IF EA=1 THEN POKE POR+2,DRAW: Y8=Y8+3:
DRAW=DRAW+16: IF DRAW>50 THEN DRAW=1.
6150 IF EA=2 THEN Y9=Y9+3
6160 GOTO 6200
6170 EA=INT(RND(0)*2+1)
6180 IF EA=1 THEN X8=INT(RND(0)*144+48):Y8=10:RETURN
6190 X9=INT(RND(0)*144+48):Y9=10:DRAW=1:RETURN
6200 POKE PLX+2,X8: POKE PLY+2,Y8: POKE PLX+3,X9: POKE PLY+3,Y9



- F. X = ML1+1 to (whatever) step 10 : F. X1 = x to x+6 : IF Peek(x1) = 95 then poke x1, 90 : ~~Poke~~ x1 = x1+1 : Poke (x1), 165
- .

- F. x = ML1 + 1 to whatever it already is step 10 : IF Peek(x) = 85 then N. x : G. wherever
- F. x1 = x to x+6 : IF Peek(x1) = 95 then poke x1, 90 : x1=x1+1 : Poke x1, 165 : N. x1 : N x2 Goto (wherever it already goes)
- ~~IF Peek (x1~~ Poke x1, 170 : N. x2
- IF Peek(x1) = 255 then Poke x1, 170 : N. x1
- N. x1 : N. x : Goto (wherever it already does)

speeded up window clearing

60 180

RND(0)*120+60

Cum
6

14
16
84
140
224

Changes

~~Prop flower pots by number~~ now - 1 +
~~RA~~

~~Change Color not long of player or girder~~

~~Change Random so pots within borders~~

~~Add number of men to score table~~

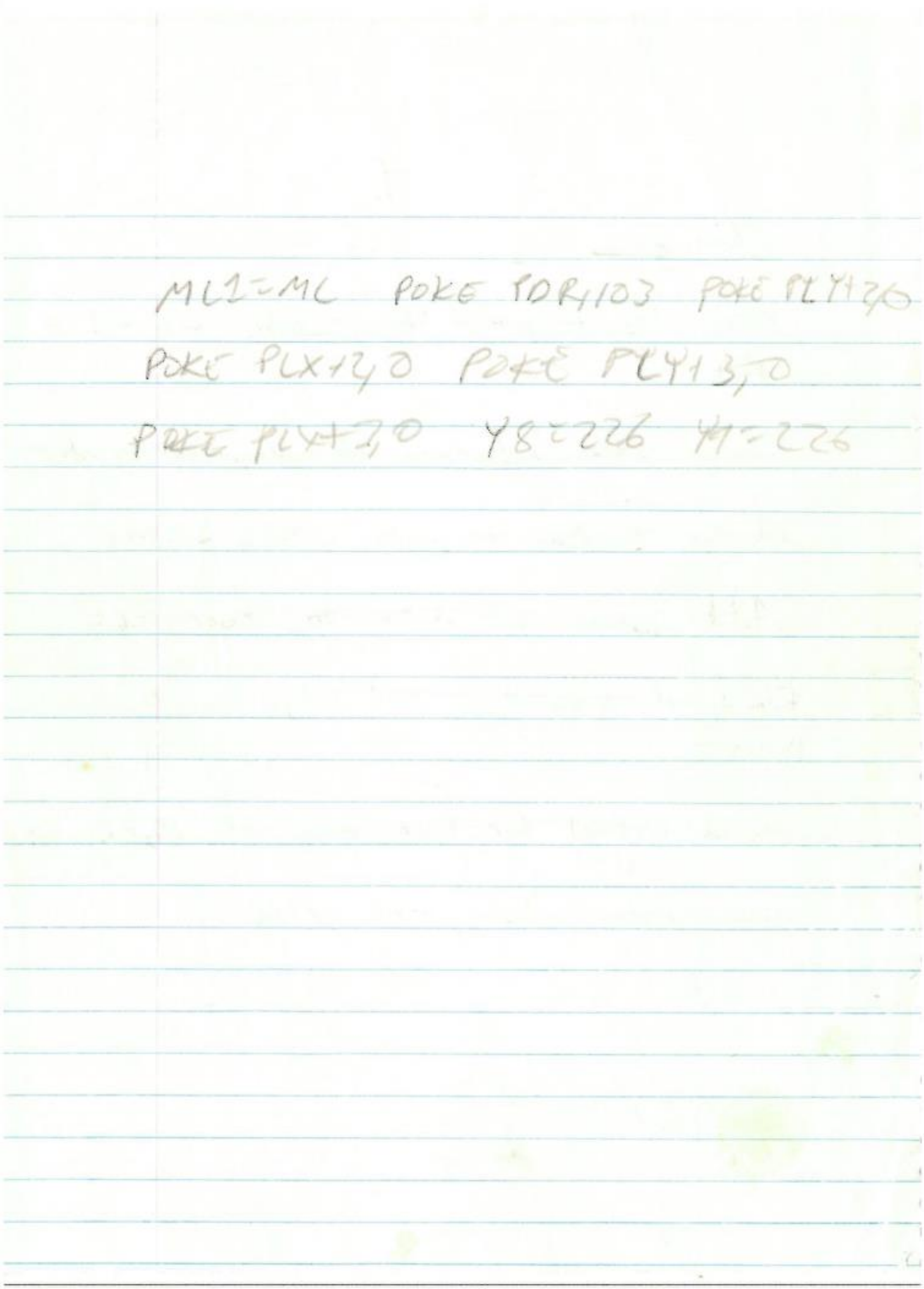
~~Flaw if you dive off~~ edge on last
~~man~~

~~Sound effect for free man at~~ 10,000 pts.
1100

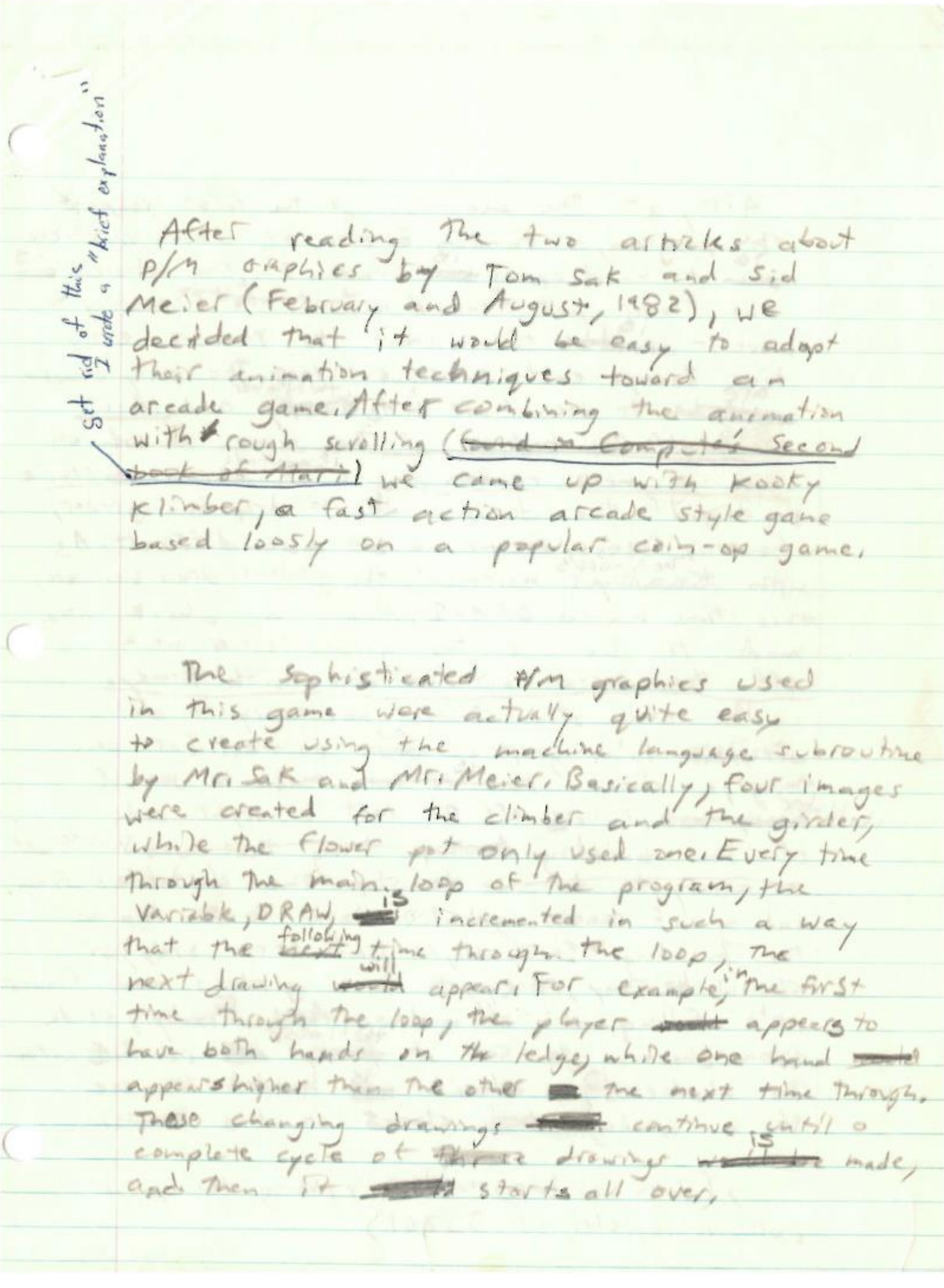
~~Make window closing~~ more often

```
FOR X=0 TO 3 POKE PCX+X,0 NEXT X
```

960



```
ML1=ML  POKE PDR,103  POKE PLY+2,0
POKE PLX+2,0  POKE PLY+3,0
POKE PLX+2,0  Y8=226  Y1=226
```



get rid of this
I wrote a "brief explanation"

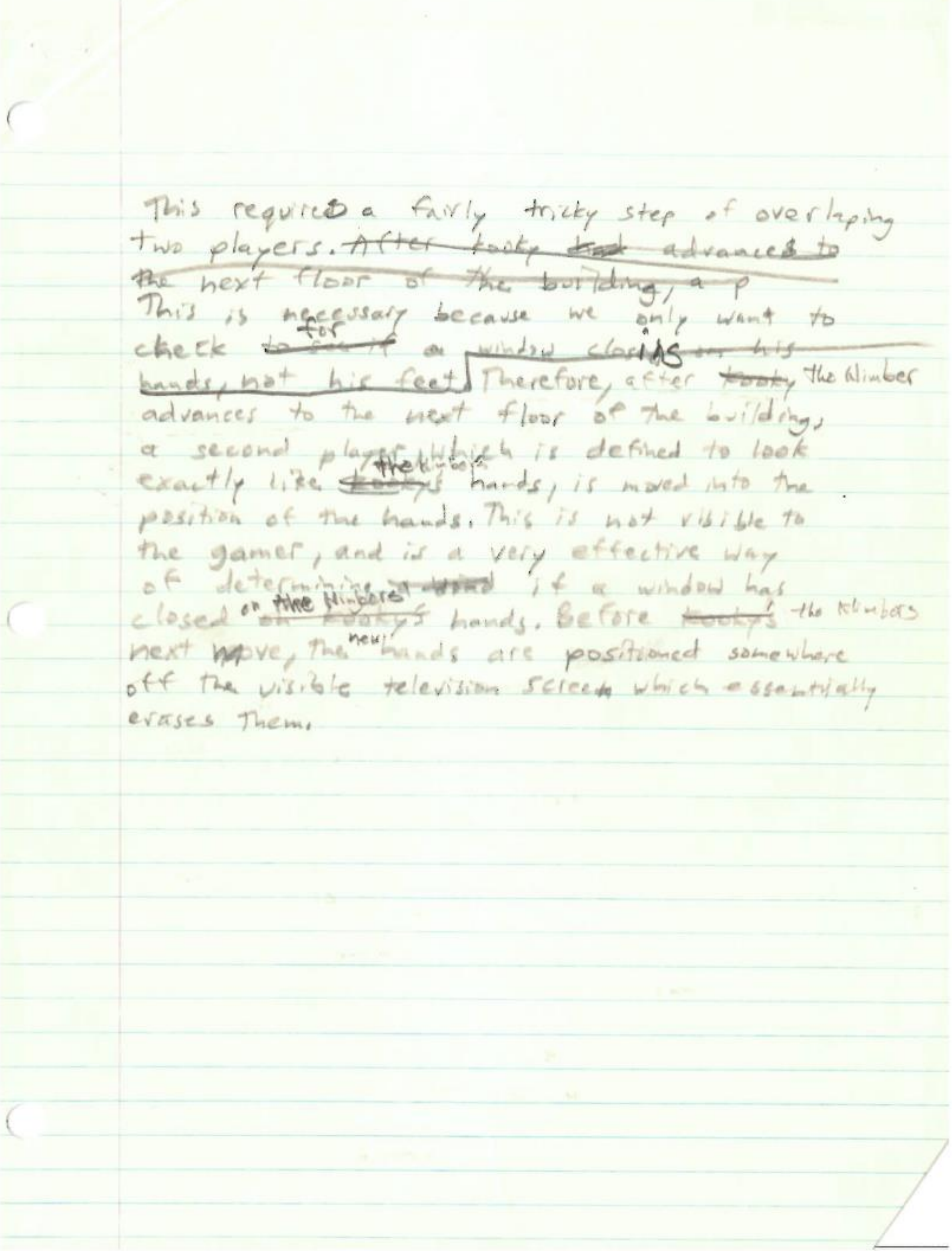
After reading The two artikles about P/M graphics by Tom Sak and Sid Meier (February and August, 1982), we decided that it would be easy to adopt their animation techniques toward an arcade game. After combining the animation with rough scrolling (~~found in Compute!'s Second book of Atari~~) we came up with Kooky Klimber, a fast action arcade style game based loosly on a popular coin-op game.

The sophisticated P/M graphics used in this game were actually quite easy to create using the machine language subroutine by Mr. Sak and Mr. Meier. Basically, four images were created for the climber and the girder, while the flower pot only used one. Every time through the main loop of the program, the variable, DRAW, is incremented in such a way that the following ~~next~~ time through the loop, the next drawing will ~~would~~ appear. For example, in the first time through the loop, the player ~~would~~ appears to have both hands on the ledge, while one hand ~~would~~ appears higher than the other the next time through. These changing drawings continue until a complete cycle of ~~these~~ drawings is made, and then it starts all over.

Also, at the beginning of the first loop of the program, a random choice of which obstacle ~~was~~ is to be dropped ~~was~~ is made. If the choice ~~was~~ is the flower pot, fine. All that ~~would be~~ is necessary to move it ~~would be~~ is to increment its Y-value every time through each subsequent loop. The only checks ~~would be~~ are to see if it hits ~~Kooky~~ the klimber and if it ~~went~~ goes off the bottom of the screen in which case a new random choice ~~was made~~ is made of which obstacle to drop. If the decision ~~was~~ is to drop the girder, however, things become a bit more difficult. As with ~~Kooky's~~ The klimber's movement, the girders draw variable, this time named DRAW1. Again a check was made to see if the girder either went off the screen or collided with kooky.

The only other things necessary to do with the P/M graphics ~~were~~ are minimal. For one thing, a check ~~had to be~~ is made to see if ~~Kooky had~~ the klimber falles off the left or right side of the building. ~~Another thing that was necessary to do was~~ It is also necessary to stop the obstacles from falling if kooky gets too close to the top of the building. After all, it would seem silly, or should we say, kooky, to have girders and flower pots falling out of mid-air. ~~And~~ Finally, aside from checking to see if ~~kooky~~ the klimber ~~had~~ collides with the obstacles, a check ~~was~~ is needed to see if a window ~~had~~ closes on his hands.

→ (which was done, incidentally, using the player 1 collision register of 53261)



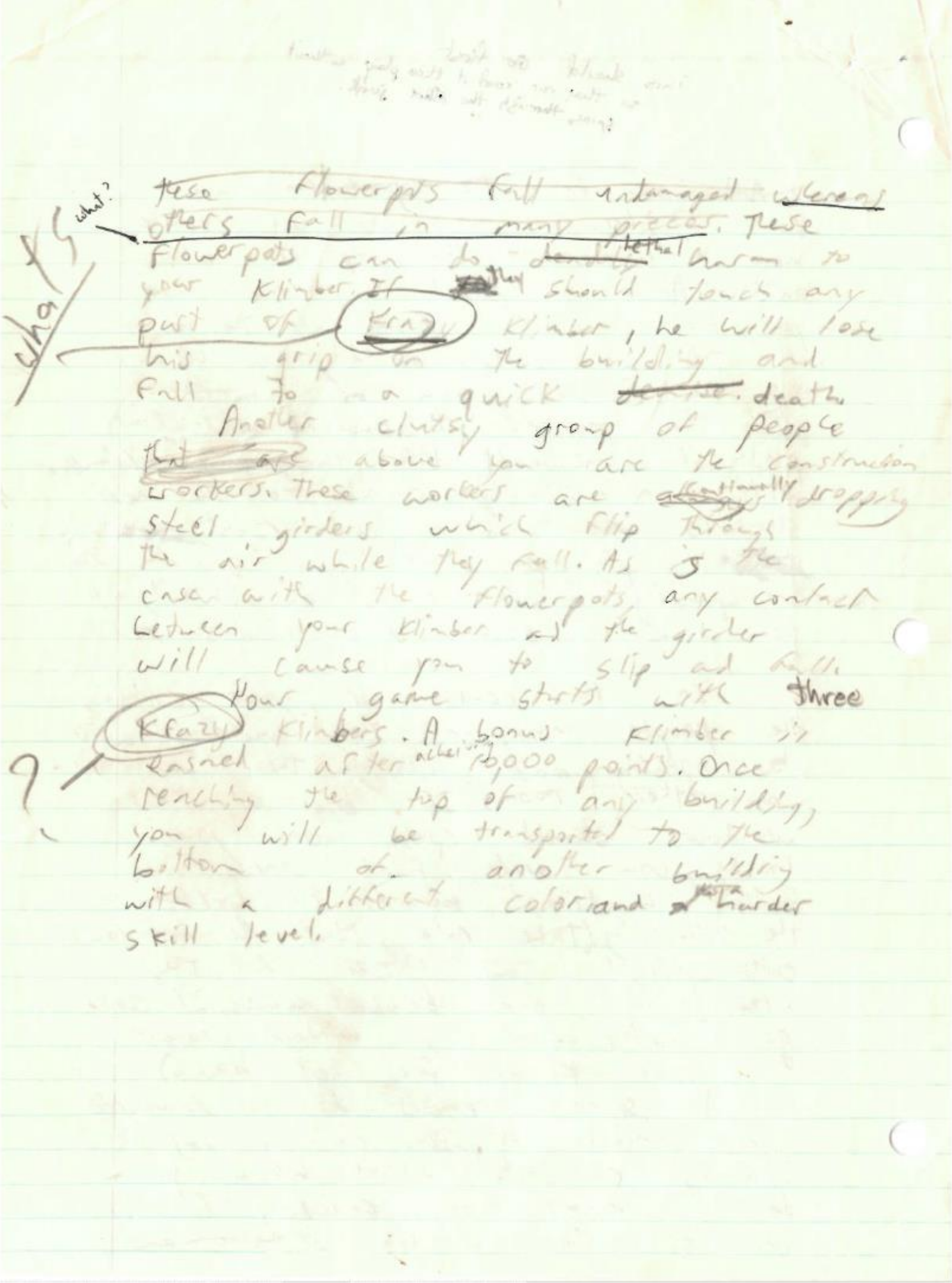
This requires a fairly tricky step of overlapping two players. ~~After Kooky has advances to the next floor of the building, a p~~ This is necessary because we only want to check ~~to see if~~ for a window closing on his hands, not his feet. Therefore, after ~~Kooky~~ the Klimber advances to the next floor of the building, a second player which is defined to look exactly like the Klimbers ~~Kooky's~~ hands, is moved into the position of the hands. This is not visible to the gamer, and is a very effective way of determining ~~a wind~~ if a window has closed on the Klimbers' ~~on Kooky's~~ hands. Before ~~Kooky's~~ the Klimbers next move, the new hands are positioned somewhere off the visible television screen which essentially erases them.

This should Go first
so they can read it then play without
going through the other junk.

~~And now for the game!~~ You are Kooky Klimber. You have an uncurvable urge to climb to the top of tall building. After a period of initialization, you find yourself poised at the bottom of the first (grey) building. To ascend ~~up~~ the building, you must first pull the joystick backwards and then push it forwards. After this, your klimber will automatically climb to the next story. To move left and right simply push the joystick in the corresponding direction. Sound easy? Well, as you will soon discover, it's not!

In the course of your climb, the people inhabiting the building will be opening and closing the windows. ~~in their~~ rooms. If one of these windows should close on your hands, you will find yourself falling headfirst towards the bottom of the building. (Take note that the windows only affect your klimber if they close on your klimber's hands. It does not matter if a window closes on your kimber's feet or body.)

A second nemisis directed towards your Klimber are the falling objects. Clumsy residents above have a deadly habit of knocking flowerpots off of their windowsills. ~~Some on~~



what's

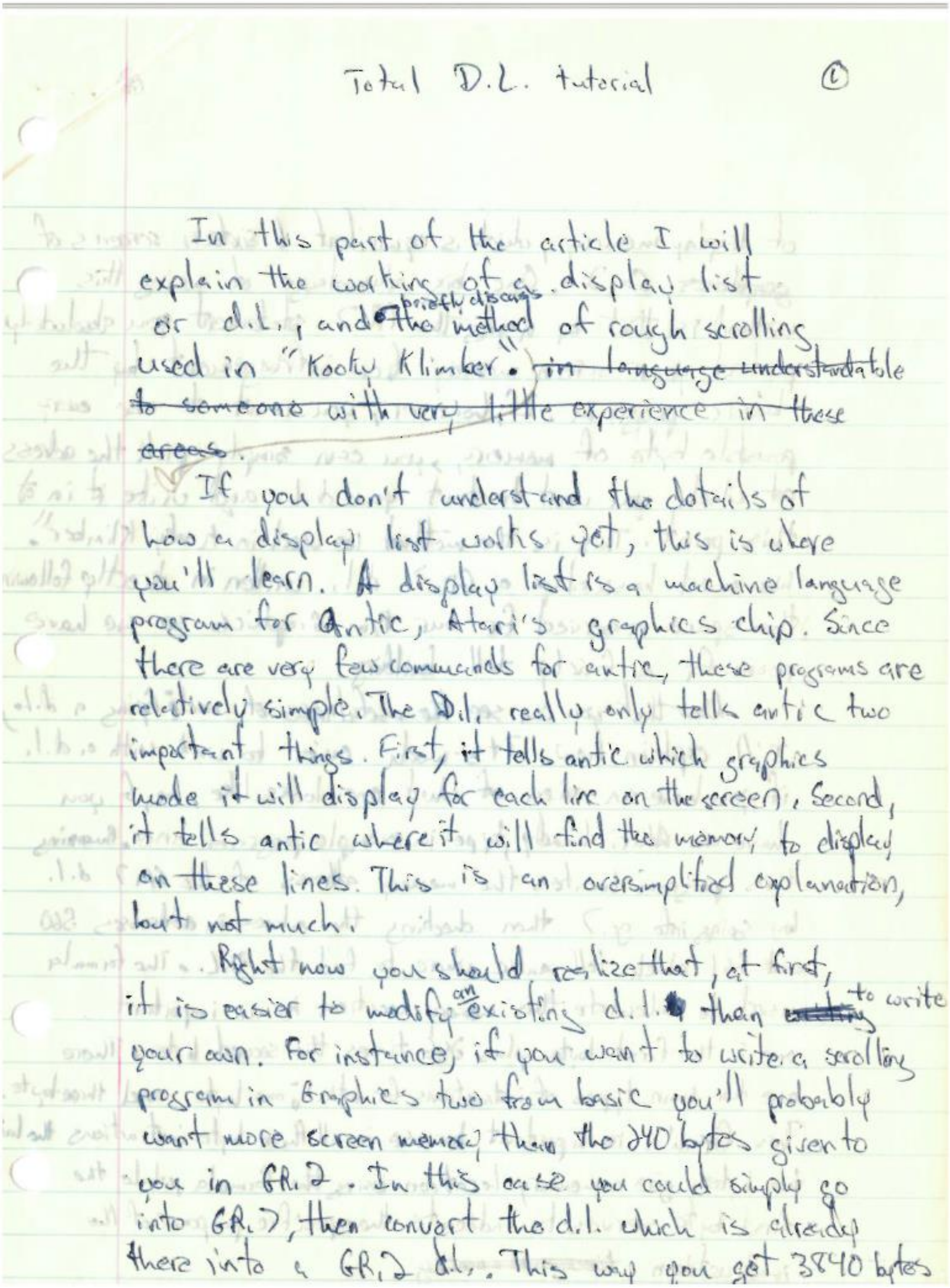
what?

~~these flowerpots fall undamaged whereas others fall in many pieces.~~ These flowerpots can do ~~deadly~~ lethal harm to your Klimber. If ~~[illegible]~~ they should touch any part of ~~Krazy~~ Klimber, he will lose his grip on the building and fall to a quick ~~demise~~. death.

Another clumsy group of people ~~that are~~ above you are the construction workers. These workers are ~~always~~ continually dropping steel girders which flip through the air while they fall. As is the case with the flowerpots any contact between your Klimber and the girder will cause you to slip and fall.

?

Your game starts with three Krazy Klimbers. A bonus Klimber is earned after achieving 10,000 points. Once reaching the top of any building, you will be transported to the bottom of another building with a different color and ~~a~~ extra harder skill level.



Total D.L. tutorial ①

In this part of the article I will explain the working of a display list or d.l., and briefly discuss the method of rough scrolling used in "Kooky Klimber". ~~in language understandable to someone with very little experience in these areas~~.

If you don't understand the details of how a display list works yet, this is where you'll learn. A display list is a machine language program for Antic, Atari's graphics chip. Since there are very few commands for antic, these programs are relatively simple. The D.l. really only tells antic two important things. First, it tells antic which graphics mode it will display for each line on the screen. Second, it tells antic where it will find the memory to display on these lines. This is an oversimplified explanation, but not much.

Right now you should realize that, at first, it is easier to modify an existing d.l. than ~~writing~~ to write your own. For instance, if you want to write a scrolling program in Graphics two from basic you'll probably want more screen memory than the 240 bytes given to you in GR.2. In this case you could simply go into GR.7, then convert the d.l. which is already there into a GR.2 d.l.. This way you get 3840 bytes of display memory, which is equivilent to sixteen screens of ~~graphics~~ GR.7. One big advantage of using this method is that by using the GR.7 statement you absolutely protect your screen memory from infringement by the basic program. If, however, you want to use every possible byte of memory, you can simply pick the adress at which you want to start your d.L. and write it in at this point. This is the method we used in "Rocky Klimber". We used basically a Gr. 3 d.l. written in directly following the space reserved for our P/m Graphics so we have room for a fairly tall building.

Now that you can see the usefulness of modifying a d.l., I'll explain how. It's much easier to work with a d.l. if you have an idea of what one looks like, so, if you have an Atari handy, type in example program one. Running this program locates the memory address of the GR.7 d.l. by going into gr.7 then checking the values in addresses 560 and 561 which tell antic where to find the D.L. The formula used to calculate this memory location is an important one: the first byte plus 256 times the second byte. There are two main types of instructions for antic; one-byte and three byte. This formula is important because in all three byte instructions the last two bytes give a memory location using the formula, while the first byte serves to indicate the specific purpose of the instruction, ~~[illegible]~~

Now we can go through and examine the d.l., instruction by instruction. The first three instructions in the d.l. are all one-byte instructions. These tell antic too leave lines blank. The formula for this type of instruction is the number of lines you wish left blank minus one times sixteen. Since these instructions are 112's on all Atari d.l.s they tell antic to leave three sets of eight lines blank right away. ~~the purpose of~~ these are used so the display won't start above the top of your T.V. screen. After these comes the first three byte instruction. This is a load memory scan or L.M.S instruction; it tells antic where to look for the memory it will use to display on the screen. The first byte of this instruction is a combination of the LMS. indicator which is 64 and a graphics 7 instruction which is 13, for a total of 77. Actually, this instruction only tells antic where to find the memory for this first line of GR.7 and it's possible to use these instructions to point out memory for each of the lines individually. Fortunately, you don't have to do this, because for each successive line after the first LMS. instruction antic just uses the memory directly following what it used for the last line. As I said before, the last two bytes of a 3-byte instruction are the ones that contain a memory pointer, so using our formula of the first byte (actually the second in a three-byte instruction) plus 256 times the second, you can calculate the start of graphics screen display memory. After the LMS instruction come a lot of thirteens. Each of these tells antic to display one line of

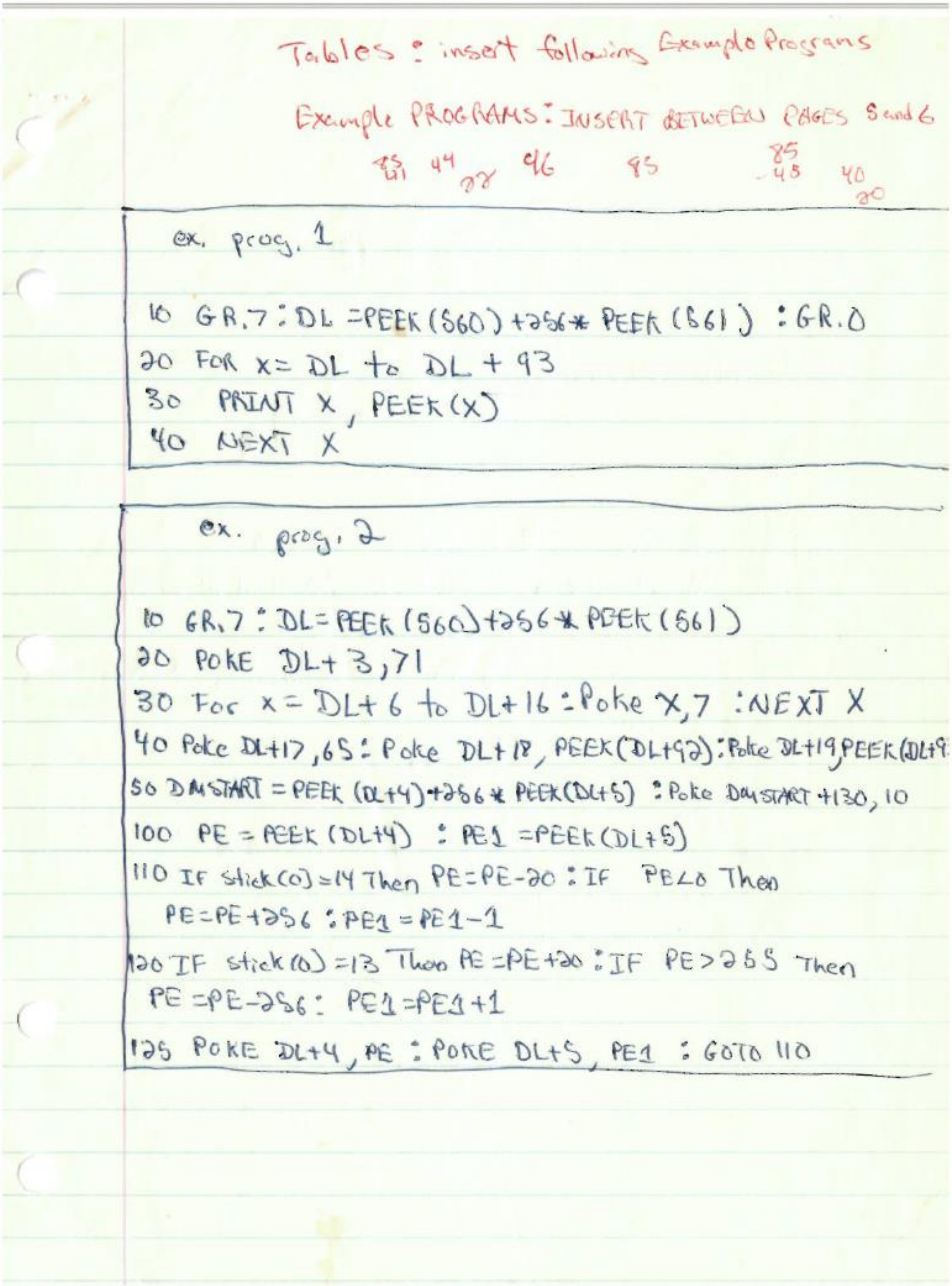
GR.7 (a complete list of these instructions is given in table one) just as the first byte of the LMS instruction did, except these don't ~~have~~ need the added 64 (L.M.S.-indicator) or memory pointers. Directly following the GR.7 instructions is another LM.S. instruction followed by three twos. This is the set of instructions for the text window, which is like a miniature display list within a display list. Next come the last three bytes of the display list. These are a jump on vertical blank or JVB instruction which tells antic the location of the start of the D.L. so it can run through the whole thing again (it does this sixty times per second). The indicator for a J.V.B. instruction is a 65, which is not added to a graphics instruction. Using the memory pointer formula again, you can see for yourself that it does indeed point to the beginning of the d.l.. The antic instructions I've mentioned so far are enough to allow you to write your own d.l., but a complete (as far as I know) list of instructions is given in table 2.

Example program two utilizes the techniques I have previously discussed by converting a GR.7 d.l. to GR. 2. It also includes a method of rough scrolling in basic. The first line of the program puts the screen into graphics seven and locates the beginning of the GR.7 d.l. The second line begins the seven to two conversion by changing the first byte of the L.M.S. instruction, which had been 77 or 64+13 to 71 or 64+7 since 7 is the antic code for Gr. 2. Line 30 converts eleven more lines to Gr. 2 for a total of twelve lines. Line 40 finishes off the conversion by poking in a J.V.B. directly following the GR. 2 instructions. Since I don't know how much memory everyone will use when typing this program in, the numbers for the pointer of the J.V.B are found by peeking ~~what had been~~ the original GR. 7 J.V.B. The fact that the rest of the old GR. 7 D.L. is still there doesn't matter since antic will never get there, jumping to the top of the D.L. each time it reaches the new J.V.B..

¶ Lines 50 through 125 give an example of simple rough scrolling. Now that you know the function of a load memory scan instruction this method of scrolling should be easy for you to understand. Since the L.M.S. instruction tells antic where to find the first byte of memory it will use to display on the screen, all we have to do is change this instruction and antic will look where we tell it, making the image on the screen appear to move. In fact, this is what lines 110, 120 and 125 do, they change the value of the pointer part of the L.M.S. by 20 (the no. of bytes in one line of Gr. 2) making the asterisk which has been poked onto the screen seem to move up and down. The variables PE and PE1 keep track of the values in the second and third bytes of the

L.M.S. instruction. If you don't understand exactly how a memory pointer works, I'll try to clarify things a little. In a normal two digit number, say 27, there is a tens digit and a ones digit, in order to calculate the value of this number you could add the ones' digit (7) to ten times the tens digit (2). If, however, you get a ones digit greater than nine you must bump up the tens digit by one and drop the ones digit back down. This is how a memory pointer works, except that instead of ones and tens your dealing with ones and 256s; the first byte is the ones and the second byte is the 256s. You can now understand why, in line 120, when PE goes above 255 we must add one to PE1 and subtract 256 from PE.

I hope this article has helped clear up some of the mysteries of the D.L. for you. The uses of modified D.L.S ar virtually limitless, and with a little experimentation you should be completely comfortable working with one.



Tables : insert following Example Programs

Example PROGRAMS : INSERT BETWEEN PAGES 5 and 6

85 41 44 28 96 85 85 -45 40 20

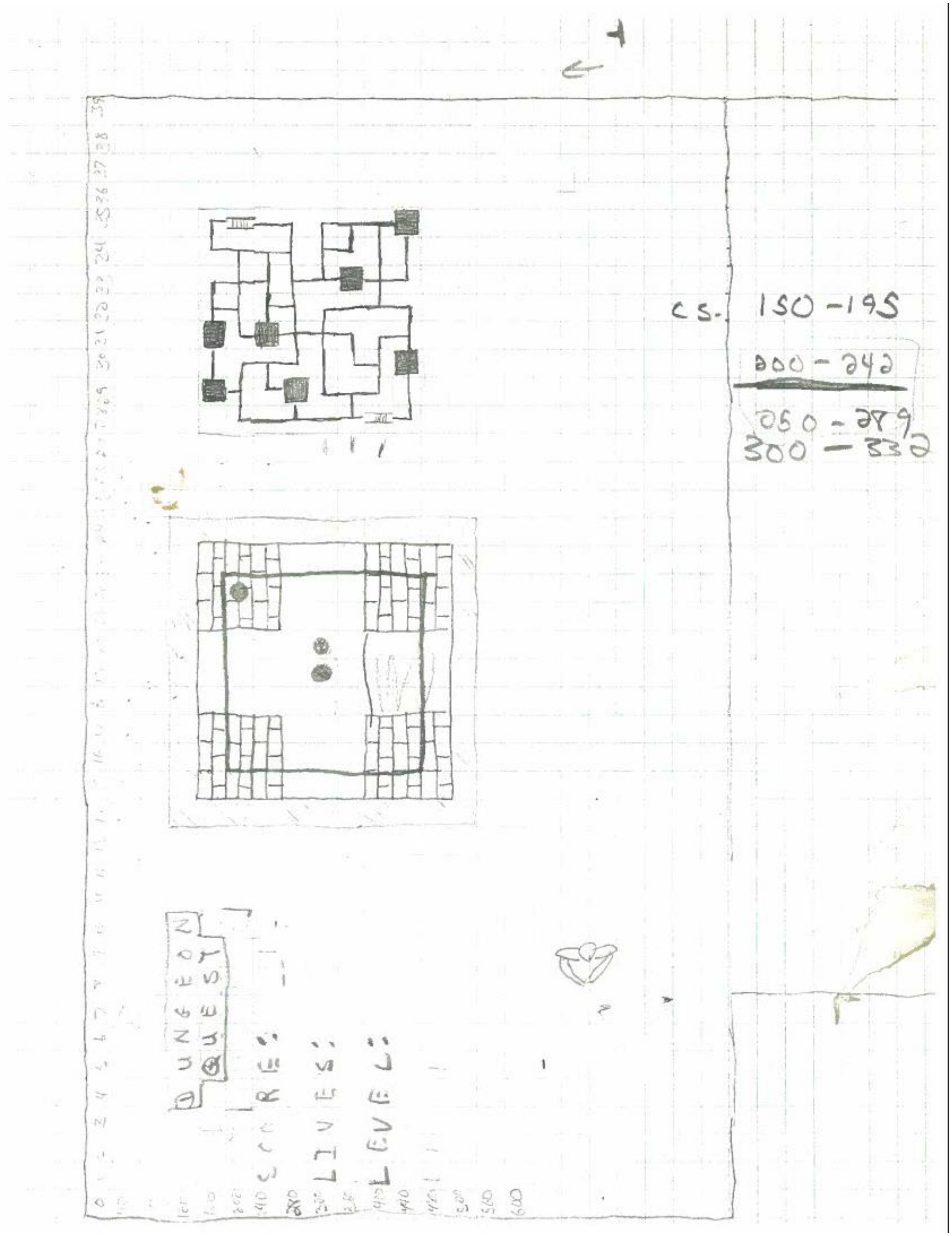
ex. prog. 1

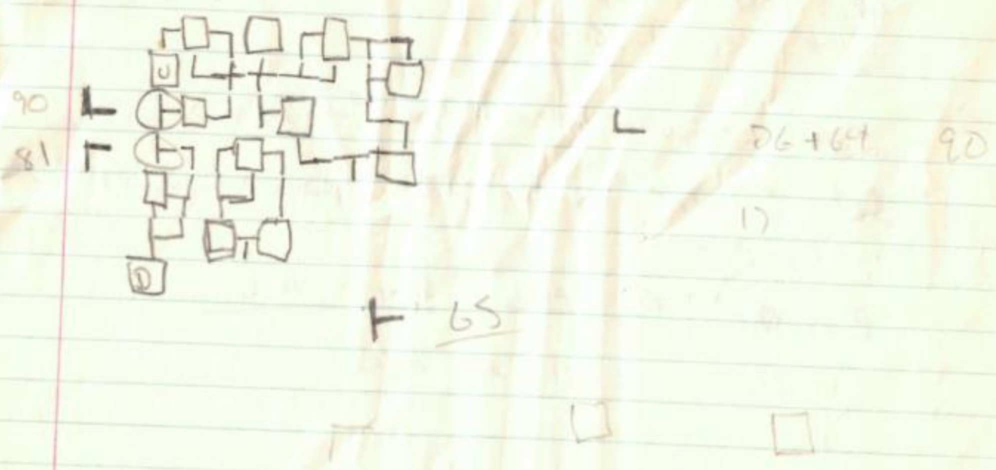
```
10 GR.7 : DL =PEEK($60) +256* PEEK($61) :GR.0
20 FOR X= DL to DL + 93
30 PRINT X , PEEK(X)
40 NEXT X
```

ex. prog. 2

```
10 GR.7 : DL= PEEK ($60)+256* PEEK ($61)
20 POKE DL+ 3,71
30 For x = DL+ 6 to DL+16 : Poke X,7 :NEXT X
40 Poke DL+17,65 : Poke DL+18, PEEK(DL+92) : Poke DL+19, PEEK(DL+9
50 DMSTART = PEEK (DL+4)+256* PEEK(DL+5) : Poke DMSTART +130, 10
100 PE = PEEK (DL+4) : PE1 =PEEK(DL+5)
110 IF stick(0)=14 Then PE=PE-20 : IF PE<0 Then
  PE=PE+256 : PE1 = PE1-1
120 IF stick(0) =13 Then PE =PE+20 : IF PE>255 Then
  PE =PE-256 : PE1 =PE1+1
125 POKE DL+4 , PE : POKE DL+5 , PE1 : GOTO 110
```

KOOKY'S QUEST





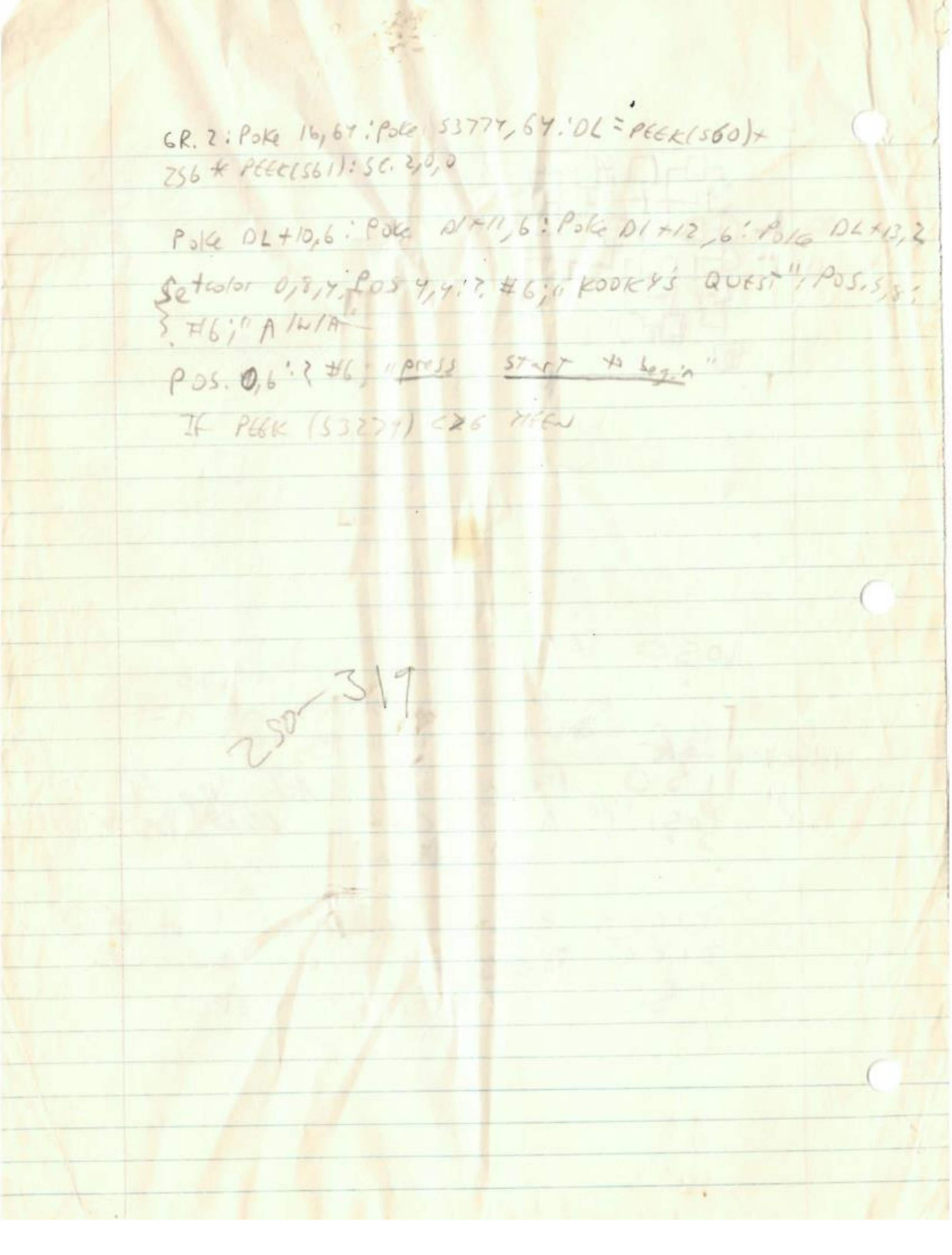
1050 After Poke 22,10 15,20 and a bunch of u=

1100 2Lines bet 1050 + 1100

1 line here → ★

1150 Fo W = A to A1 Step St :
Pos; 15, W ; ? "<<< --- <<<" Pos 15, W+B
? " " N.W

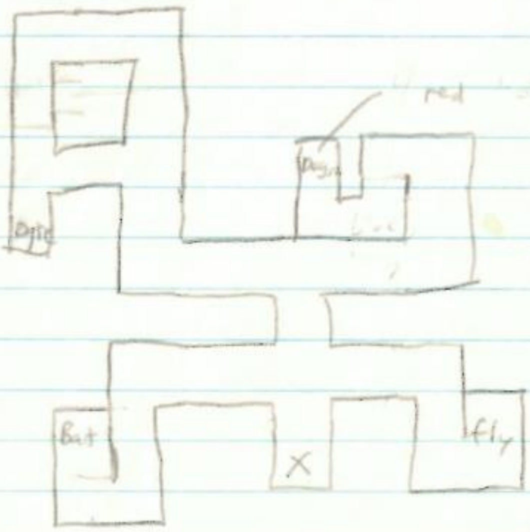
2050 same as 1050 but 2000's
2100 ~~2050~~ + 2 Lines 1 Line bet.
2150 1 Lin bet. 2100 + 2150



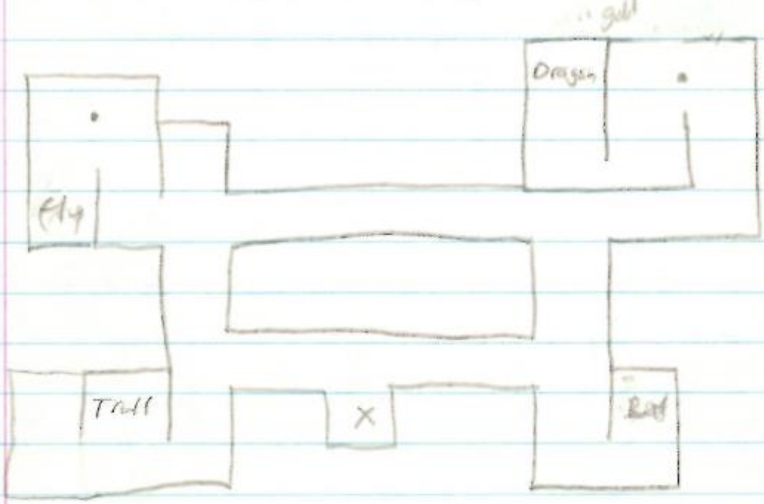
```
GR. 2 : Poke 16,64 : Poke 53774,64 : DL = PEEK(560)+
256 * PEEK(561) : SC. 2,0,0

Poke DL+10,6 : Poke DL+11,6 : Poke DL+12,6 : Poke DL+13,2

Setcolor 0,8,4 : Pos 4,4 : ? #6;"KOOKY'S QUEST" : POS.5,8 :
? #6;"A IWIA"
Pos. 0,6 : ? #6; "press START to begin"
 If PEEK (53279) <>6 THEN
```

2 50 - 3 1 9

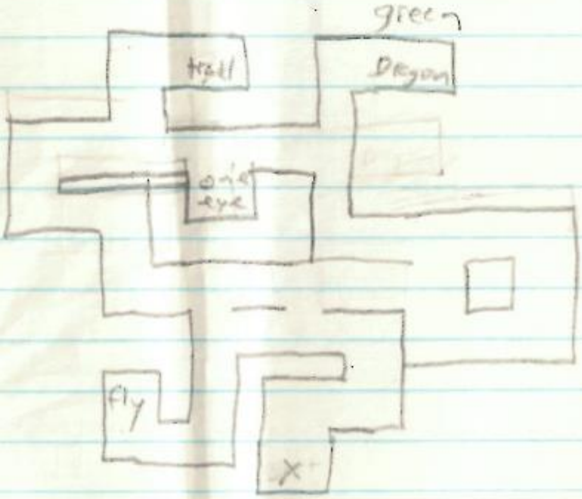
LEVEL 1



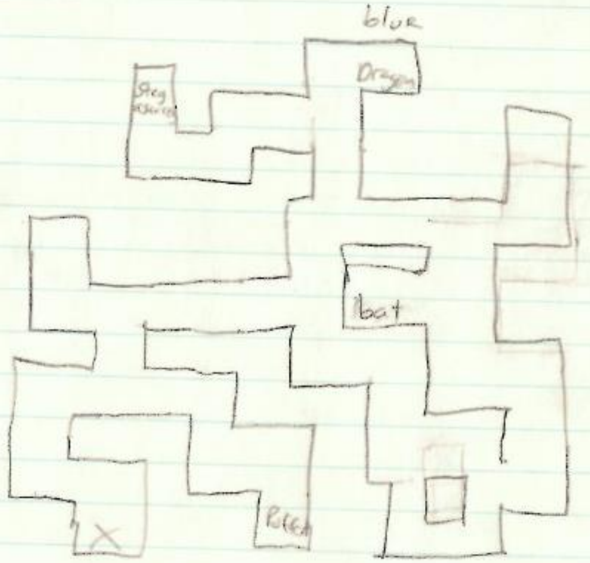
Level 2



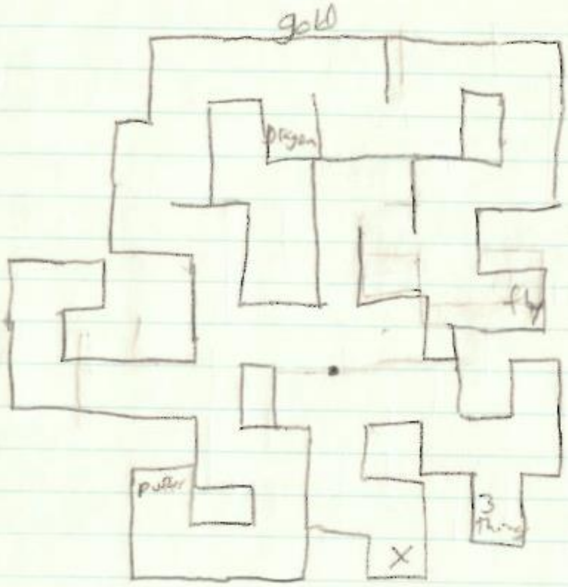
Level 3



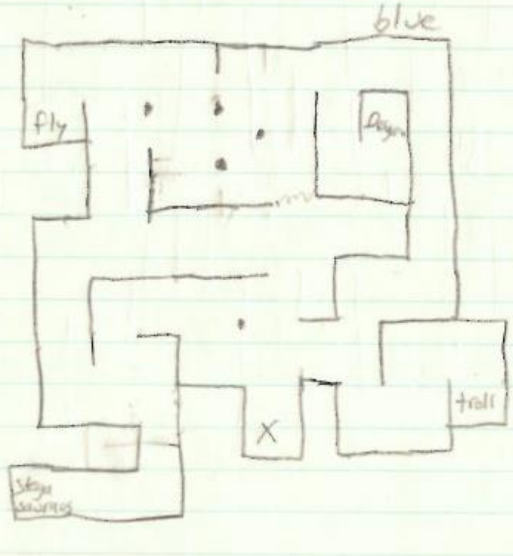
Level 4



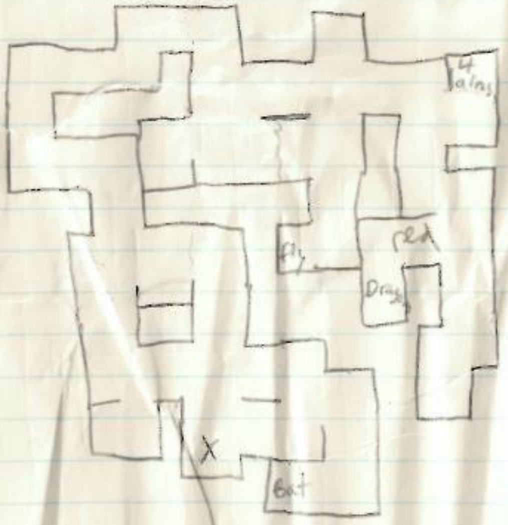
Level 5



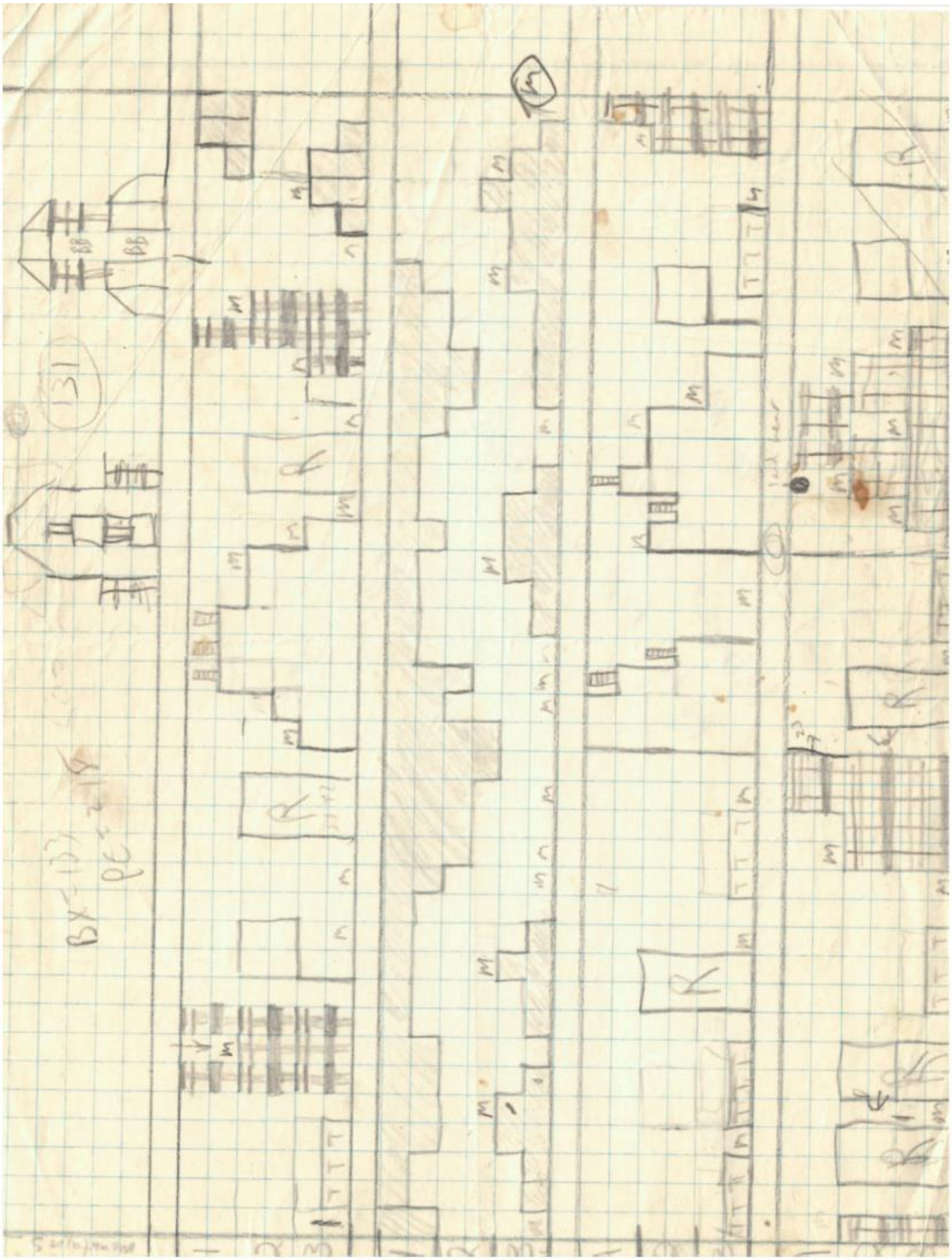
Level 6



Level 7



NIGHT RESCUE



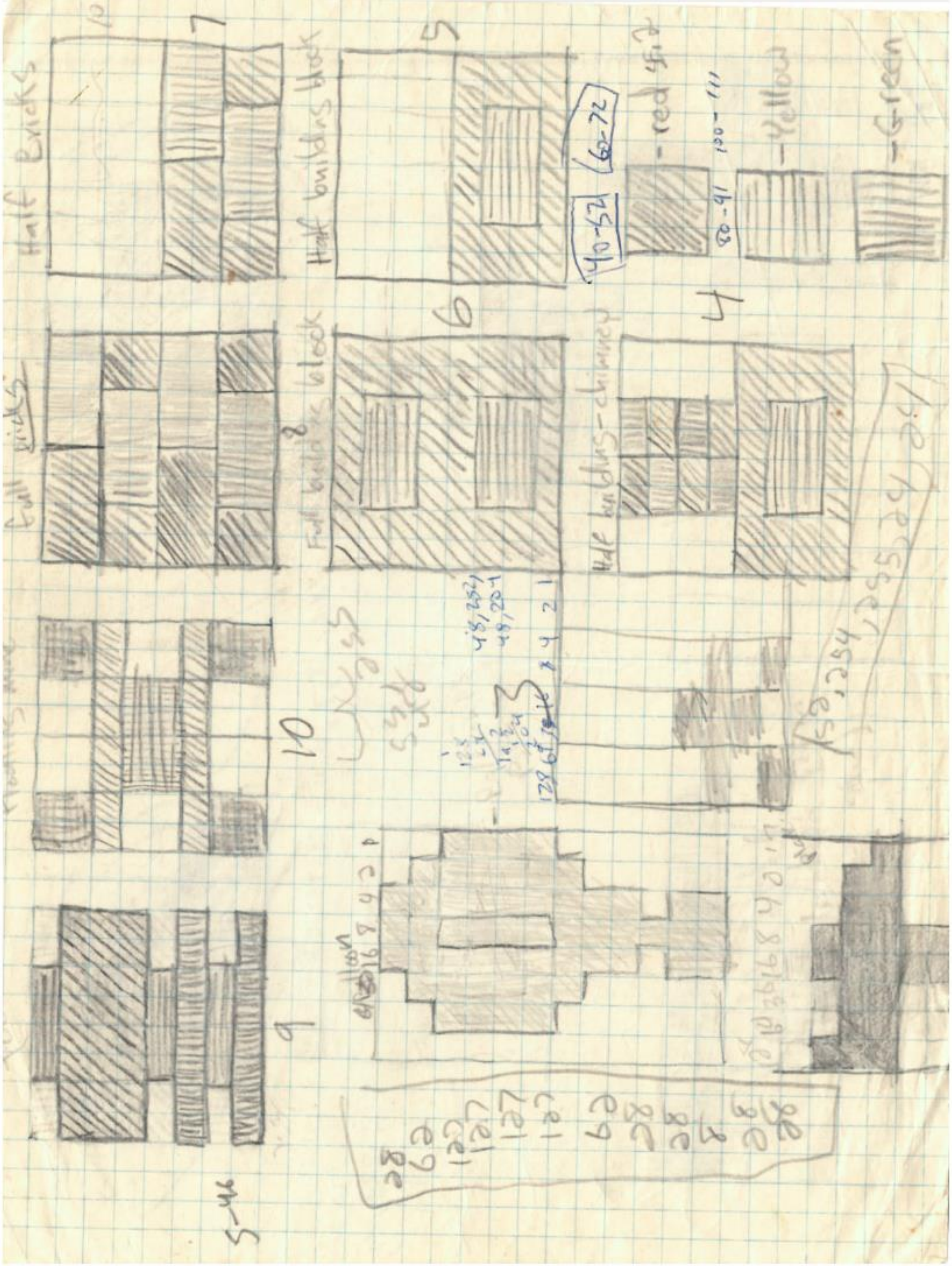
Set Color of Man

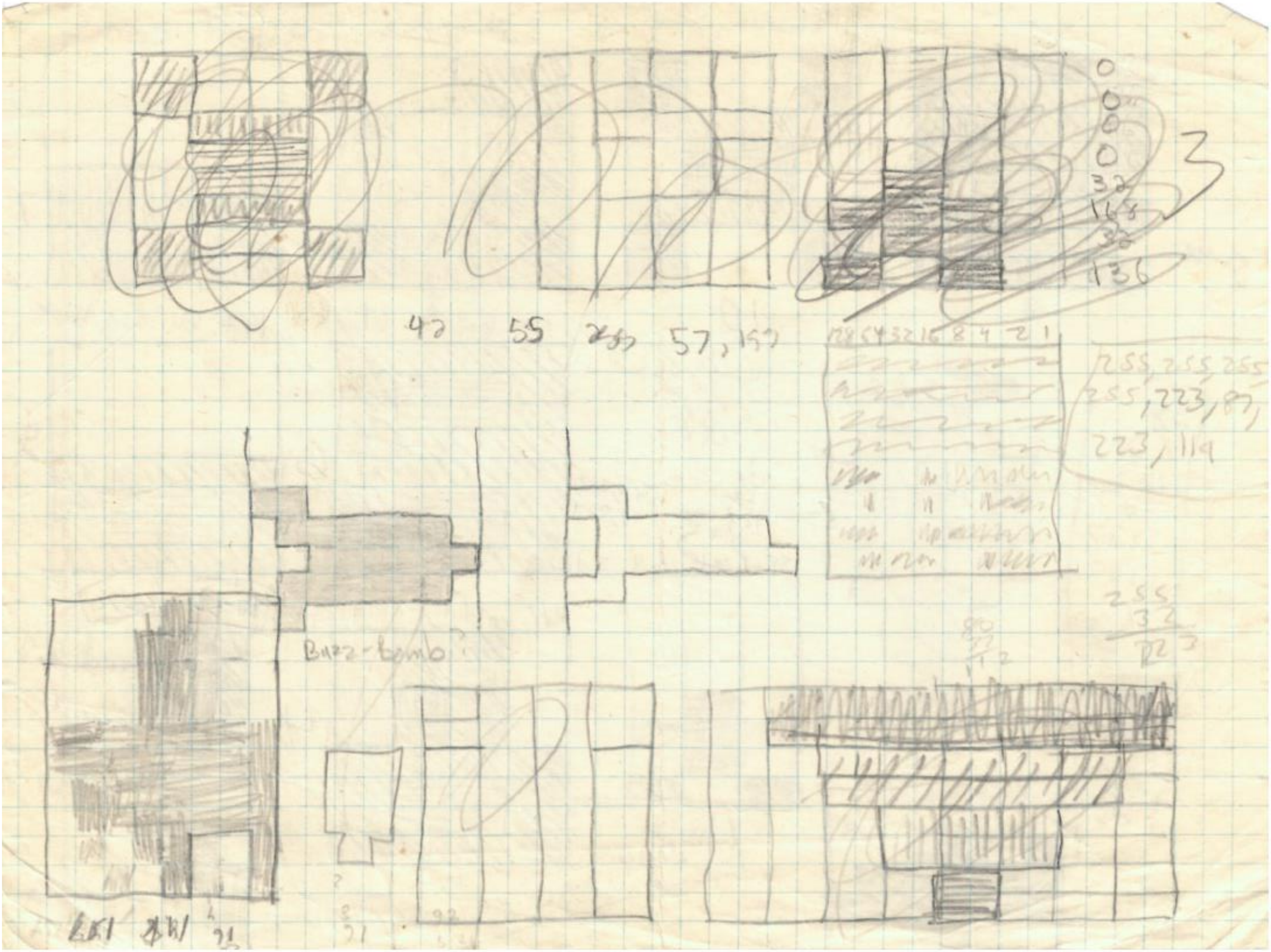
Helecopter centerly)

Position for Night issue

Move Copter down [illegible]

96.67


Half Bricks
7
Half building block
5
40-52
60-72
red
80-91
100-111
Yellow
Green
4
Half building-chimney
6
Full building block
8
Full bricks
10
9
5-46


0
0
0
0
32
168
32
136
42
55
57, 157
128 64 32 16 8 4 2 1
255, 255, 255
255, 223, 87,
223, 119
255
32
223
Buzz-bomb?

CSAVE 25-70
90-121

```
PY=PY+(PY<30 or PY>96)): Poke PL+3,PY:
PY=  3*RA+PY+ SK*RA+PY
IF (RA<30 GT
 IF RA>96) then RA=-RA
run 131
RA=RA*(-1*(RA<30 or RA>96))
INT (RND(0)*30)+99
Poke PL+3
```

36, 38

130

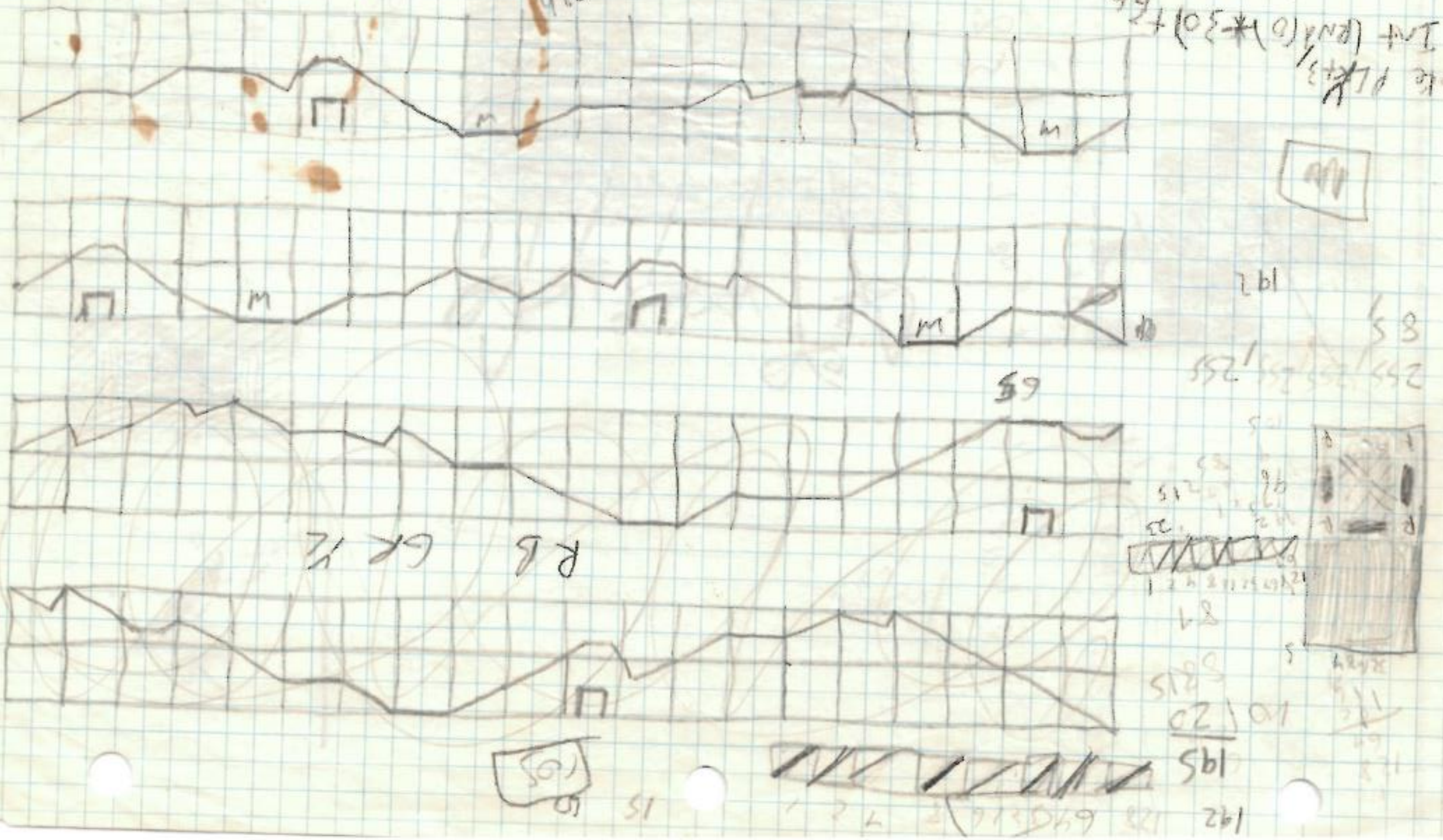
75

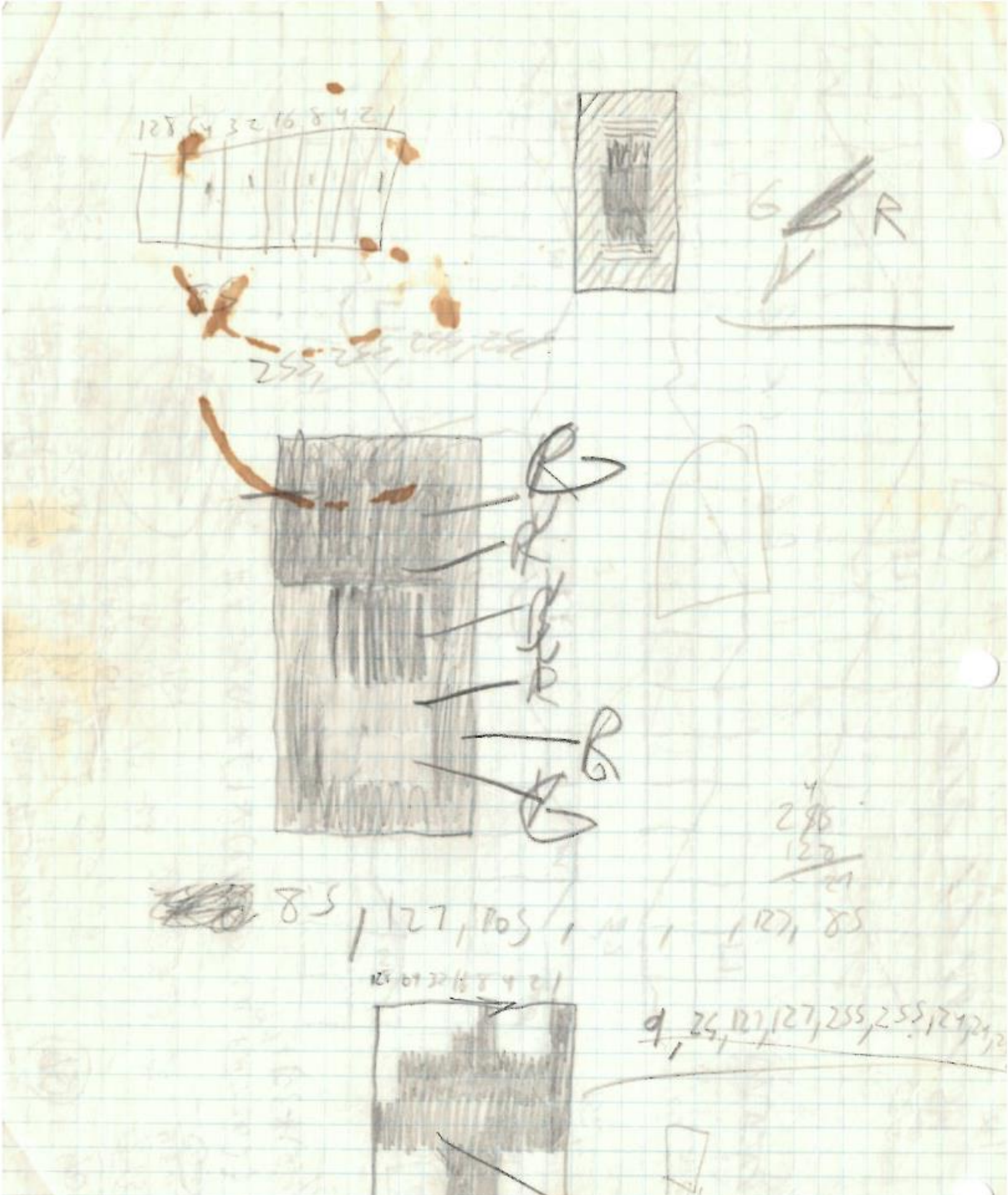
40+

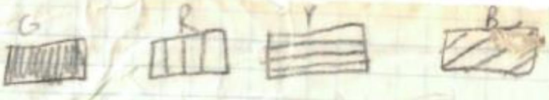
245
480
225

upper right 35

160
3
460




128 64 32 16 8 4 2 1
255, 255, 255, 255
G R
R
R
R
R
R
G
G
85, 127, 105, 1, 1, 1, 127, 85
255
127
128 64 32 16 8 4 2 1
9, 26, 127, 127, 255, 255, 124, 24, 12

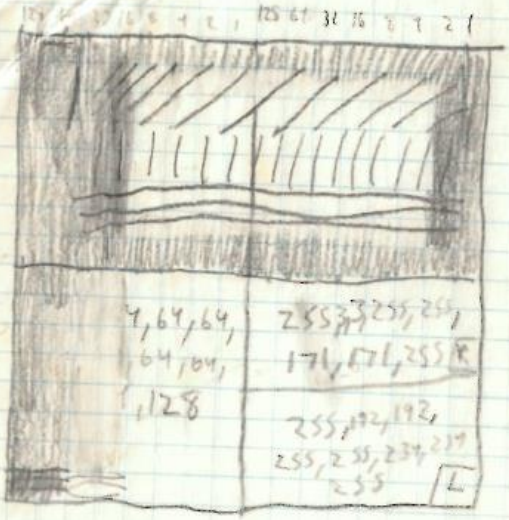


64 96

224

GR R B Y L G N

85,64,64,127,
127,106,106,
85



253,
85,1,1,253,169,169,85

64

255
64

255
128
127

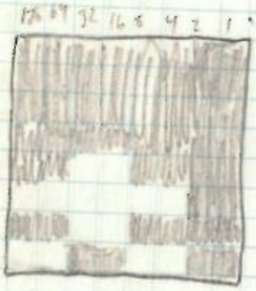
520

530

255
64
191

128
64
192

80
80
80
80
80
80
80
80
160



255,255,255,255,231,3,211,51

0,000 32 168 32 136

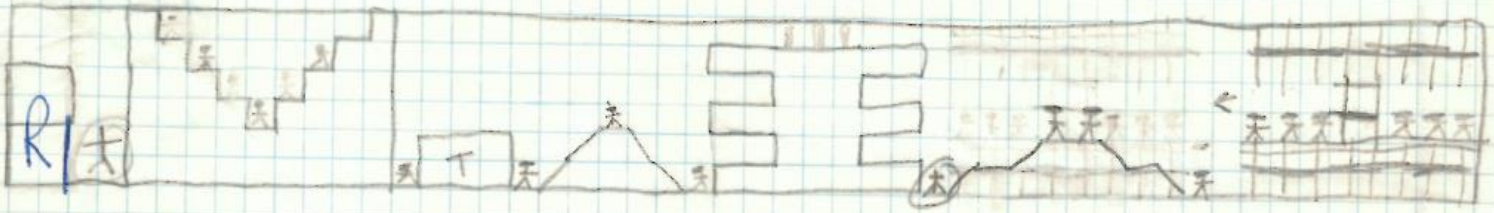
7 192 5,64

7 128 5,64

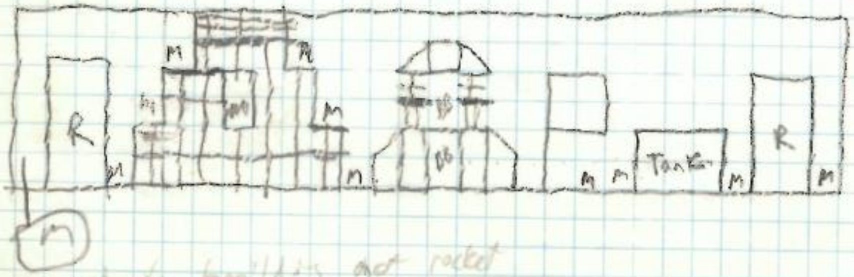
170,2,2,254,254,
170,170,170 or 0

85, 85, 85, 100, 100, 90, 90, 85
81 95

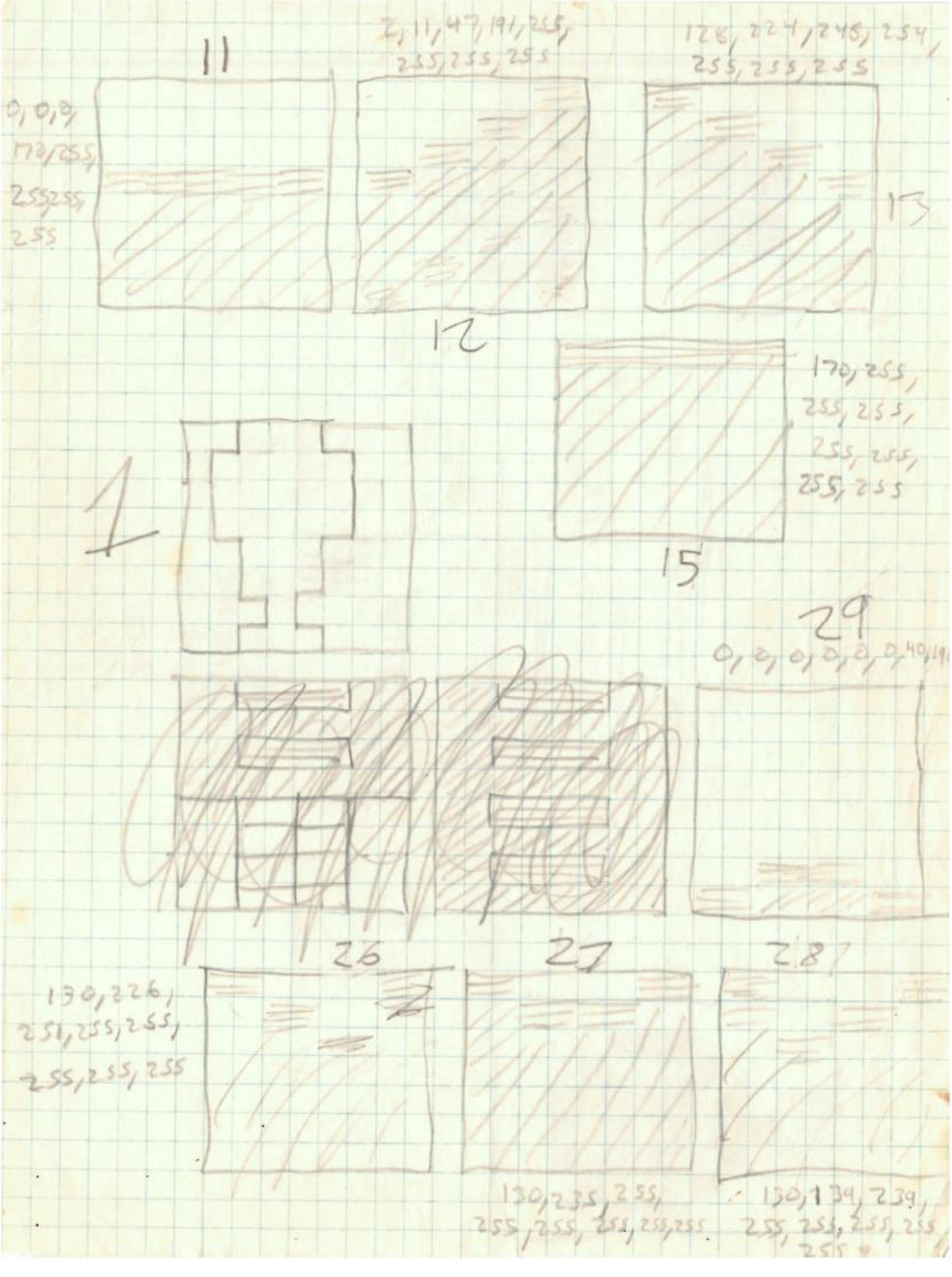
170,128,128,191,191,170,170,170 or 128

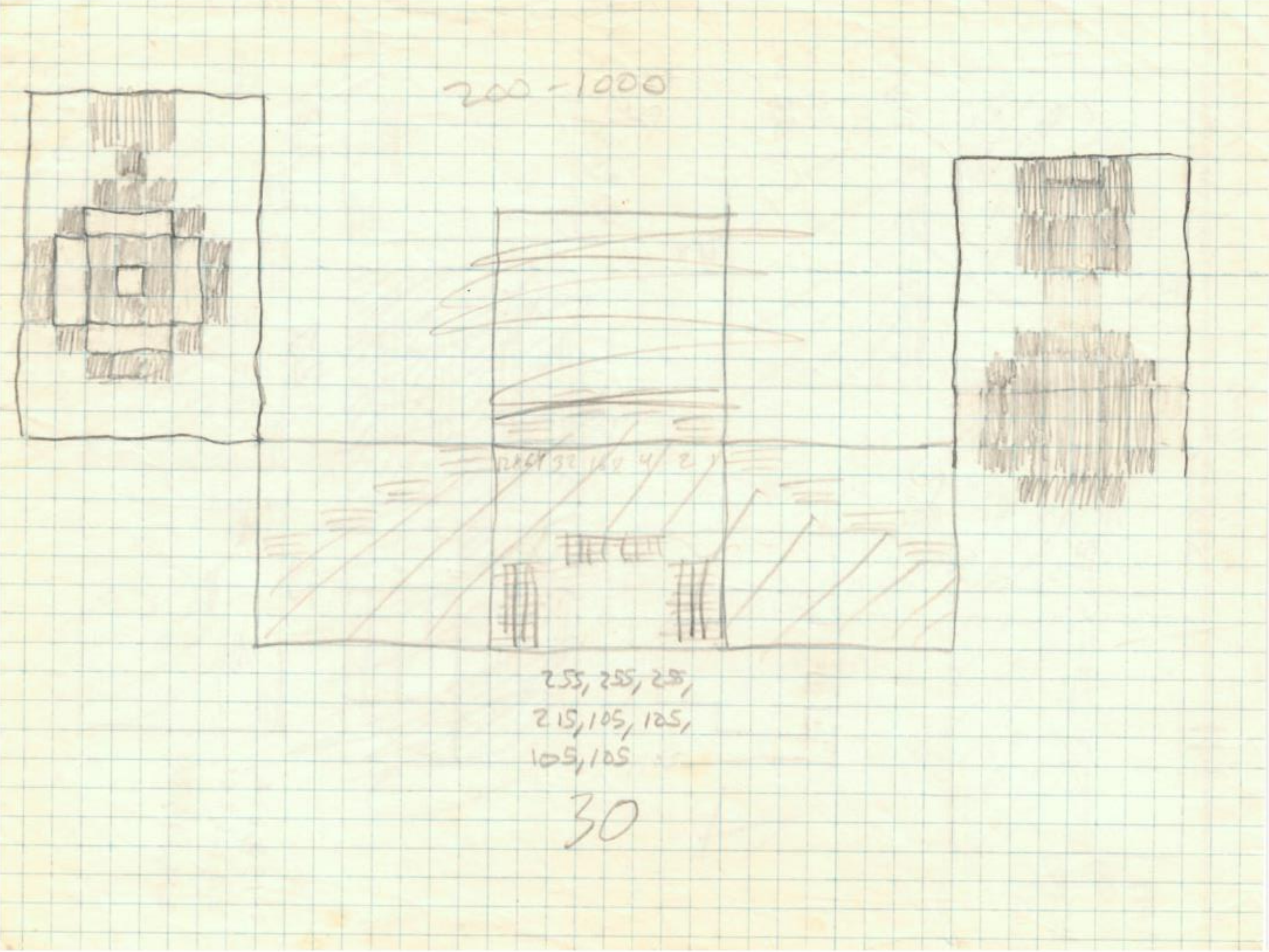


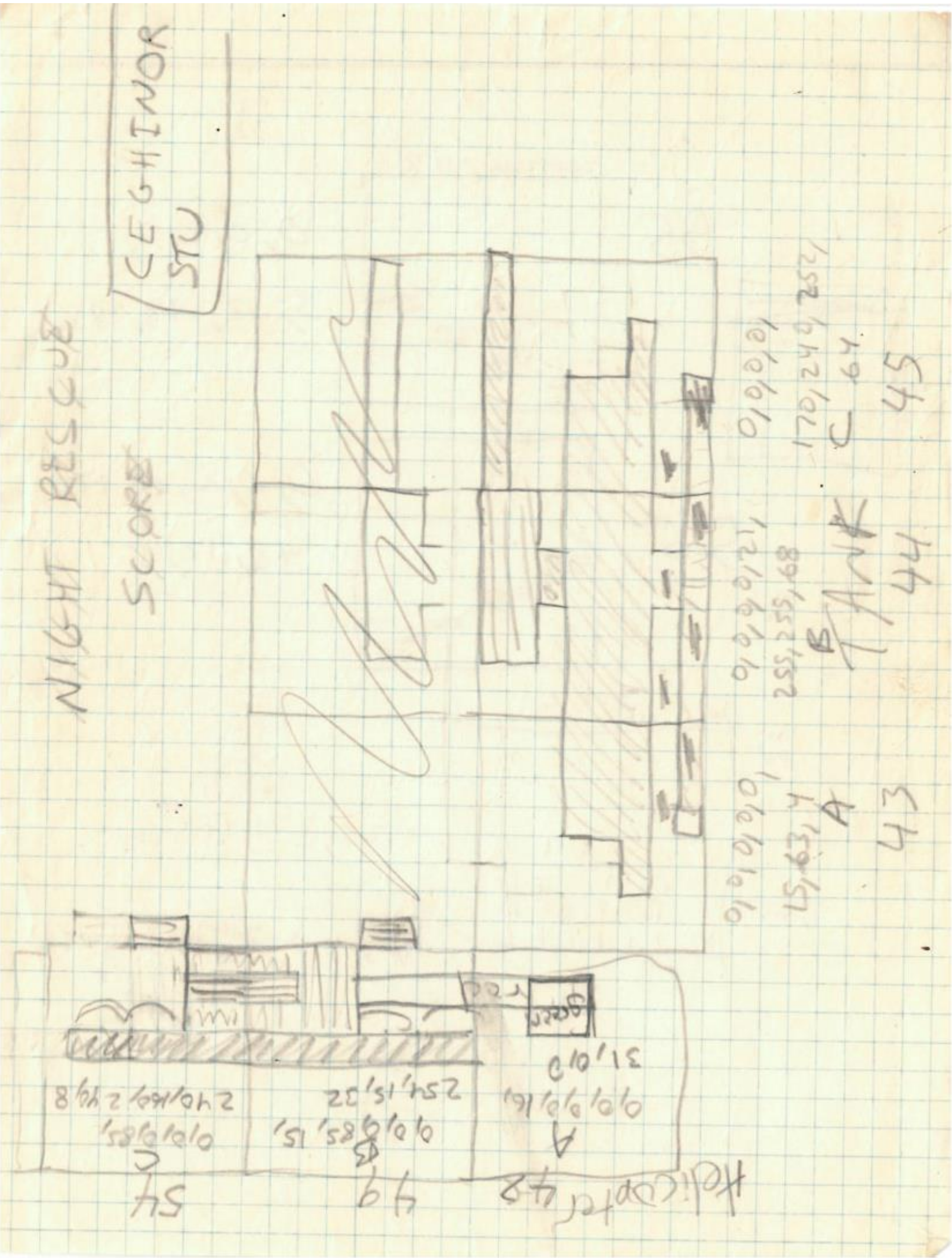
new terrain

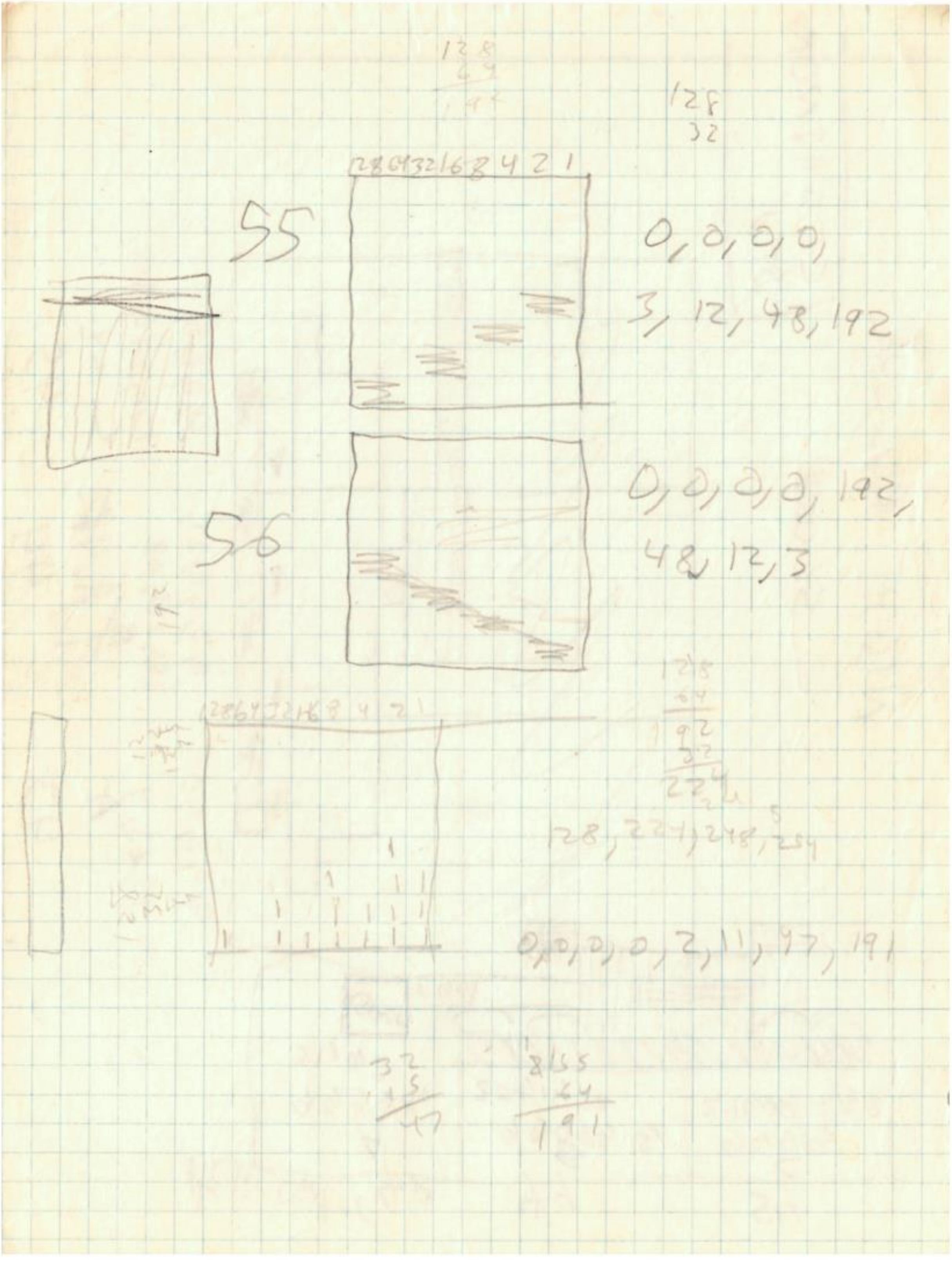


next to building not rocket

11
0, 0, 0,
170, 255,
255, 255,
255
2, 11, 47, 191, 255,
255, 255, 255
128, 224, 248, 254,
255, 255, 255
13
12
170, 255,
255, 255,
255, 255,
255, 255
15
1
29
0, 0, 0, 0, 0, 0, 40, 191
26
27
28
2
130, 226,
251, 255, 255,
255, 255, 255
130, 235, 255,
255, 255, 255, 255, 255
130, 139, 239,
255, 255, 255, 255
255


200-1000
128 64 32 16 8 4 2 1
255, 255, 255,
215, 105, 105,
105, 105
30


NIGHT RESCUE
SCORE
CEGHINOR
STU
0,0,0,0,0
12,12,0,0,0
0,0,0,0,0
170,240,252
C 64
45
255,255,68
B
TANK
44
0,0,0,0,0
15,63,4
A
43
green
red
31,0,0
0,0,0,16,
A
Helicopter 48
254,15,32
0,0,0,85,15,
B
49
240,160,248
0,0,0,85,
C
54



128
64
1 92

128
32

0, 0, 0, 0,
3, 12, 48, 192

0, 0, 0, 0, 192,
48, 12, 3

128
64
192
32
224

128, 224, 248, 254

0, 0, 0, 0, 2, 11, 97, 191

32
15
47

8155
64
191

15326

DL = 15710
13360

350

128

15360

128
40
64
3
235

56

x		x		x		x	
128	64	32	16	8	4	2	1
x	x		x		x		

GREEN

yellow

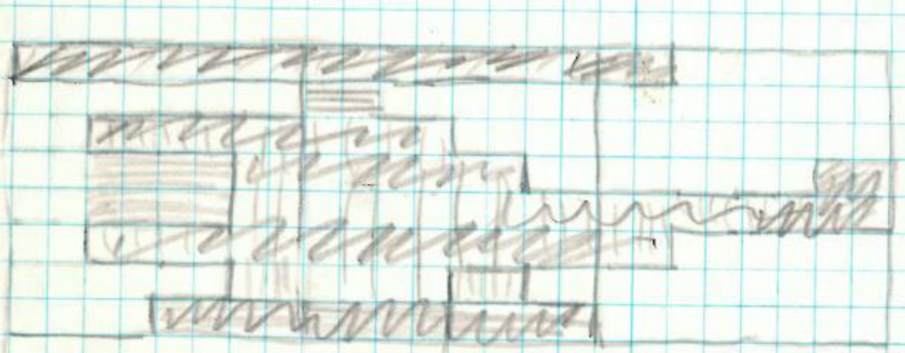
255

255, 0, 21, 41, 41, 21, 1, 10

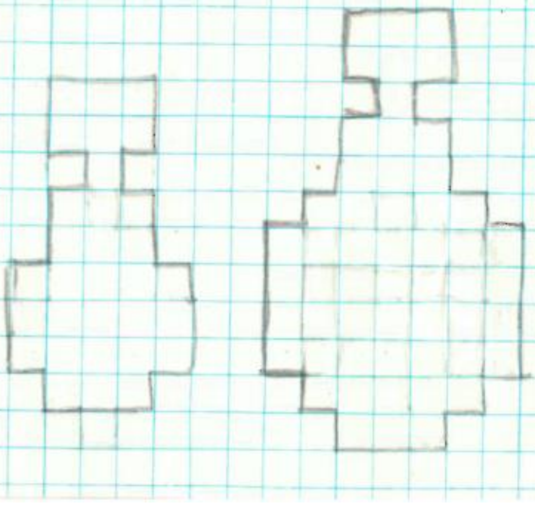
255, 188, 80, 84, 85, 85, 84, 170

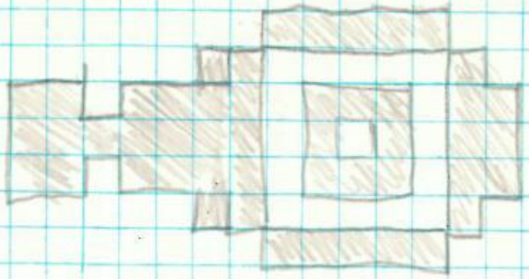
192, 0, 0, 3, 87, 64, 0, 0

85
40
85
40
15360
60
255
255
255
255
40

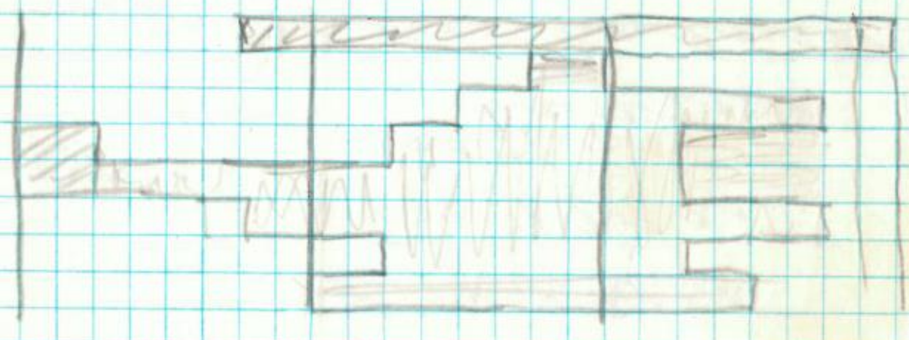


15360



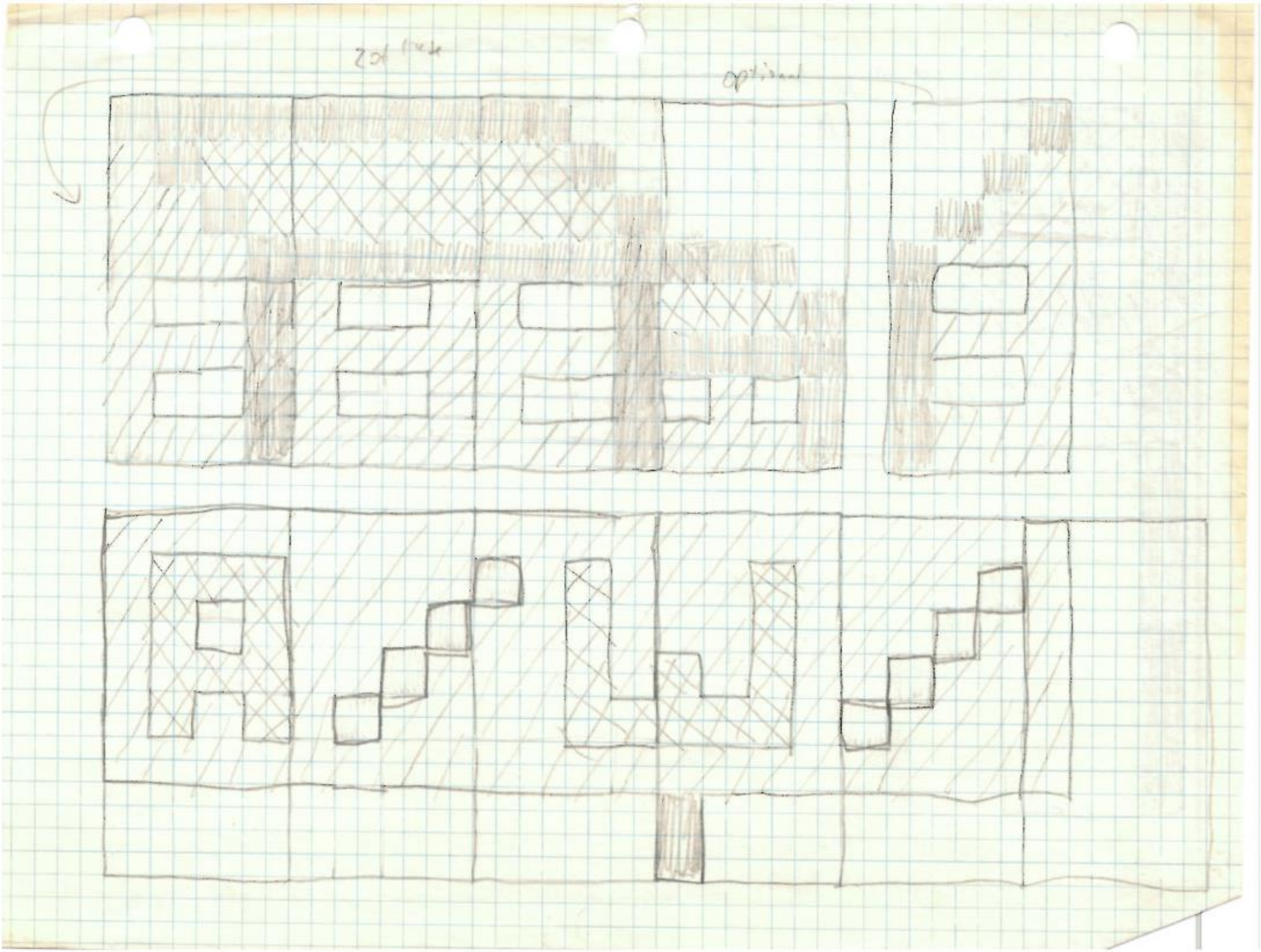


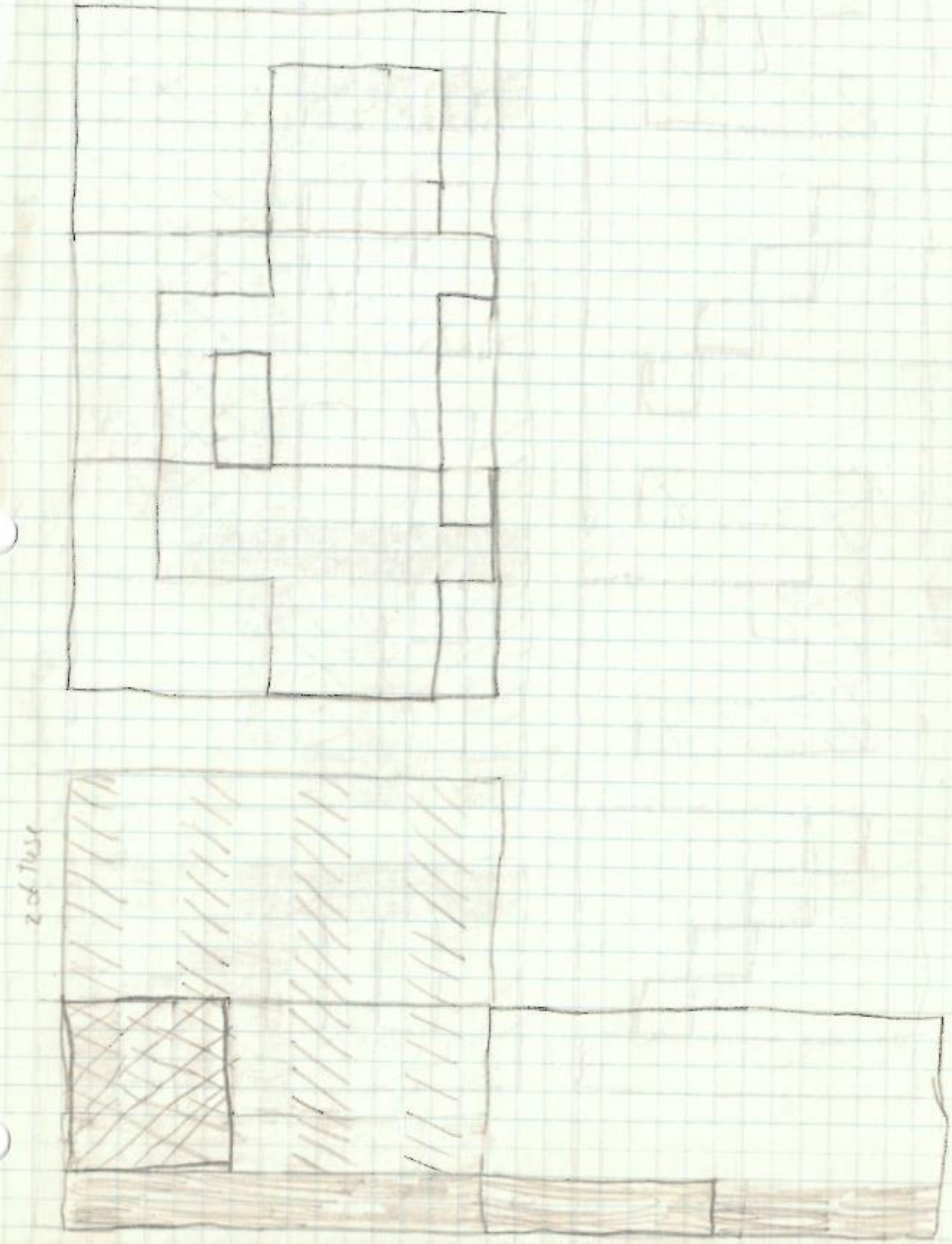
32
511
543

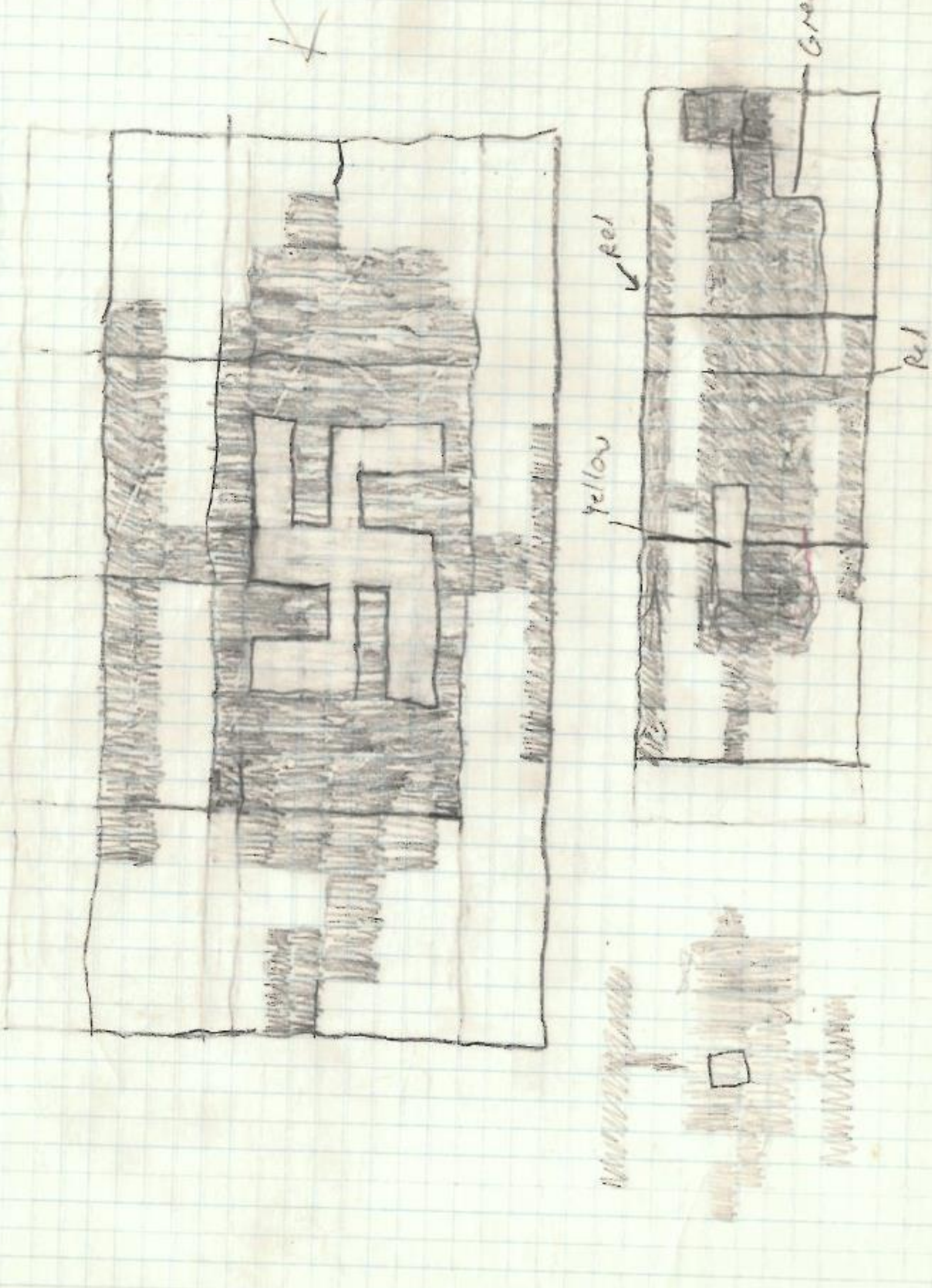


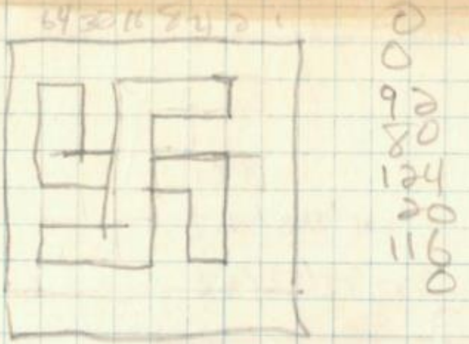
3, 0, 0, 192, 213, 1, 0, 0
255, 2, 5, 21, 85, 85, 21,
170

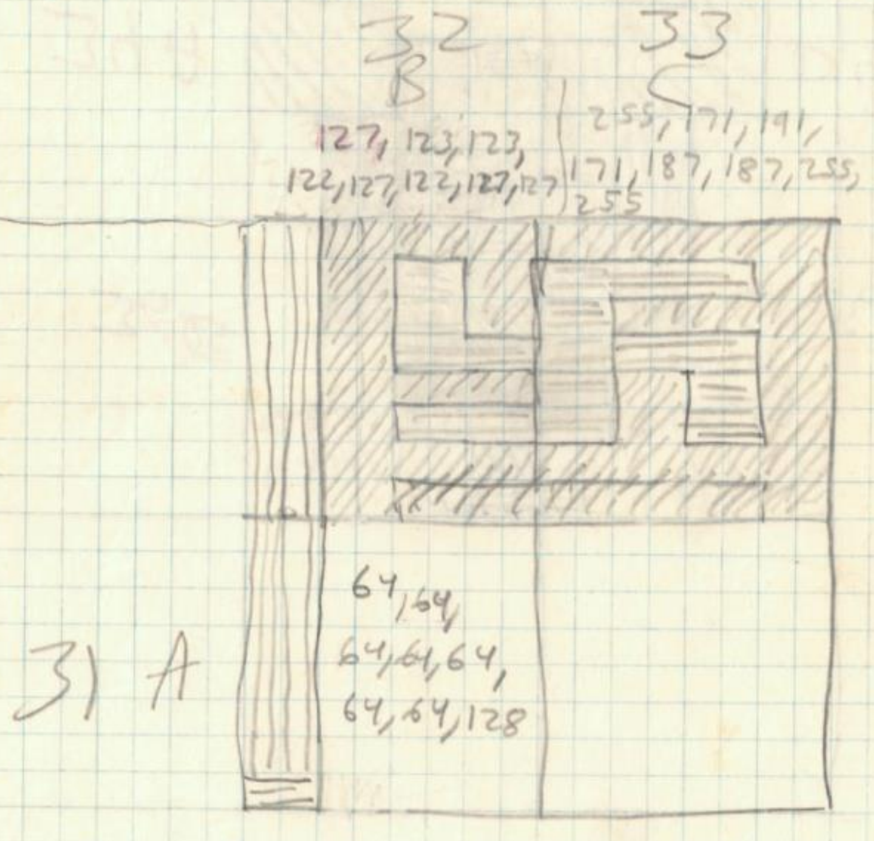
255, 0, 84, 104, 104, 84, 54,
160


2nd line
optional

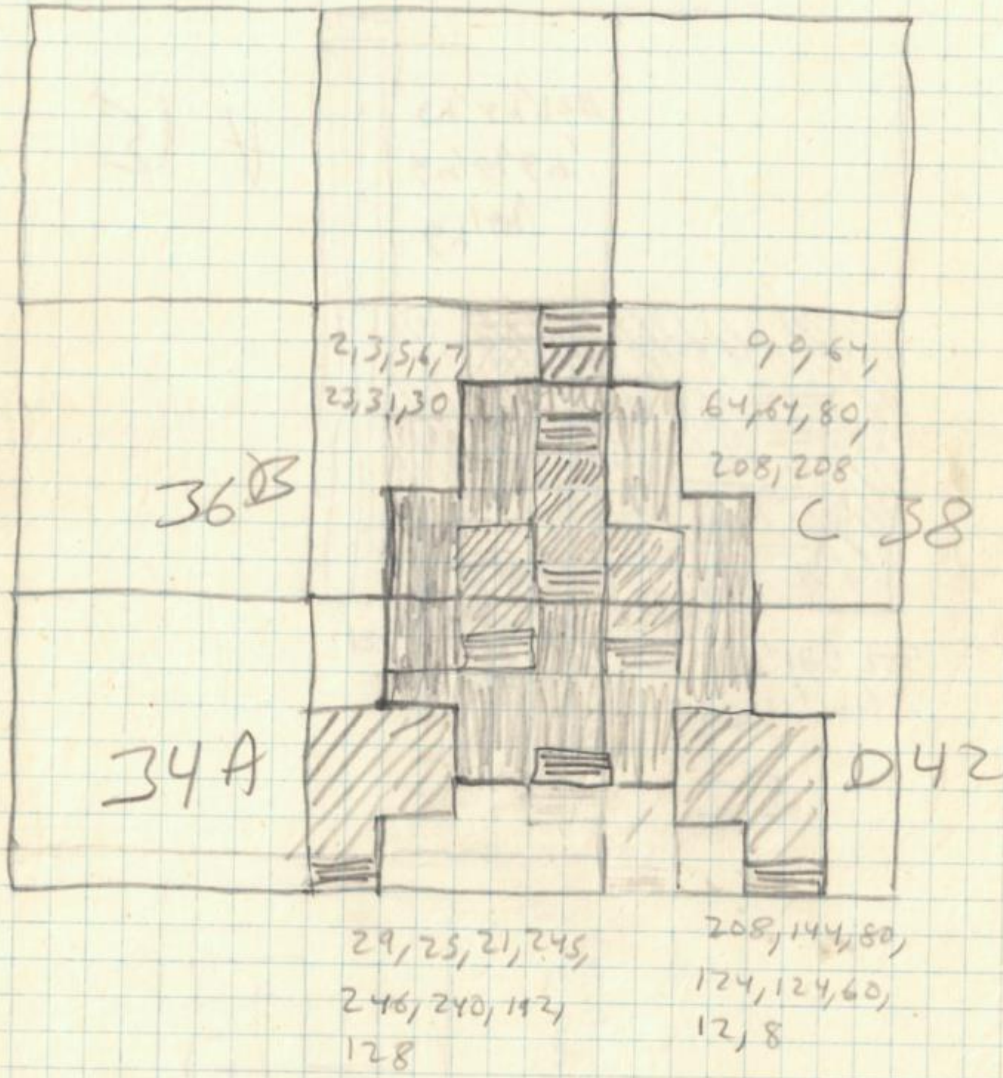

2d Tuse


Green
Red
Red
Yellow



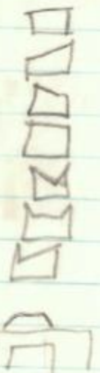


5-77
80- 142
10-82


2,3,5,6,7
23,31,30
9,9,64,
64,64,80,
208,208
36B
C 38
34A
D42
29,25,21,245,
246,240,142,
128
208,144,80,
124,124,60,
12,8

255., 255, 255, 255

215 / 1055, 105, 215

1	!	Balloon	
2	"	Reserved for future use	
3	#	Man	
4	$	Half-building chimney	
5	%	Half building block	
6	"and"	Full building block	
7	'	Half bricks	
8	(	full bricks	
9	)	tower	
10	~~x~~	~~mine~~ half bricks on top	
11	+		
12	,		
13	–		
15	/		
26	:		
27	;		
28	<		
29	=		
30	>		
31	?	FLAG A	stick
32	@	FLAG B	Left
33	A	FLAG C	Right
34	B	Rocket A	Lower left
36	D	Rocket B	upper left
38	F	Rocket C	upper Right
42	J	Rocket D	Lower Right
43	K	TANK A	
44	L	TANK B	
45	M	TANK C	
48	P	HELICOPTER A	
49	Q	HELICOPTER B	
54	V	HELICOPTER C	

55 W ◁

56 X ▷

~~57 Y~~

(12)

14636 - 14845

14852 - 15101

15108 - 15357

34, 42

36, 34

42, 131

12860

542

+1

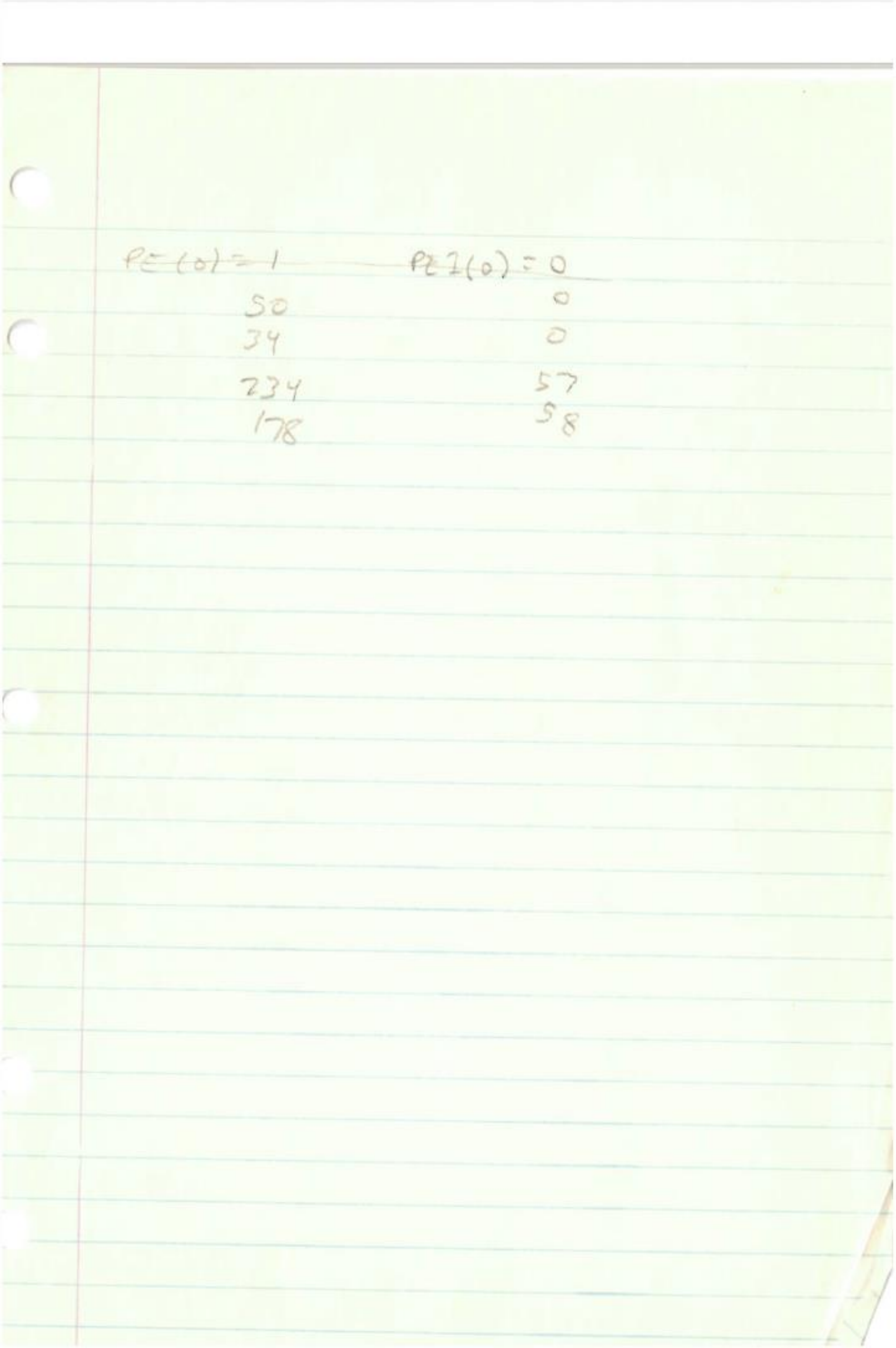
112, 112, 112, 112, 112, 66, 50, 50, 112, 112, 68, 0,
56, 112, 112, 112, 69, 0, 57, 69, 0, 58, 69, 0, 59

96

1285

DL + 11 14336 14530
17
20 15363
59 23
256

U



2 40

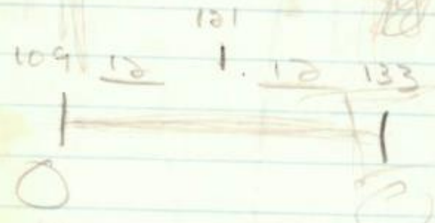
⊚

```
SM- PEEK 88+ 256*PEEK( 89 )
LOE = SM + (INT ((YP- 39)/8)+1) + 40
```

```
LINE= INT((PLY-39)/8)+1
COL = INT ((XP- 49) /8) +1
SM = PEEK(88)+ 256* PEEK(89)
LOC =SM +LINE* 40+COL-40
 PRINT PEEK LOC
```

48

$$\begin{array}{r} 183 \\ -16 \\ \hline \end{array}$$



24

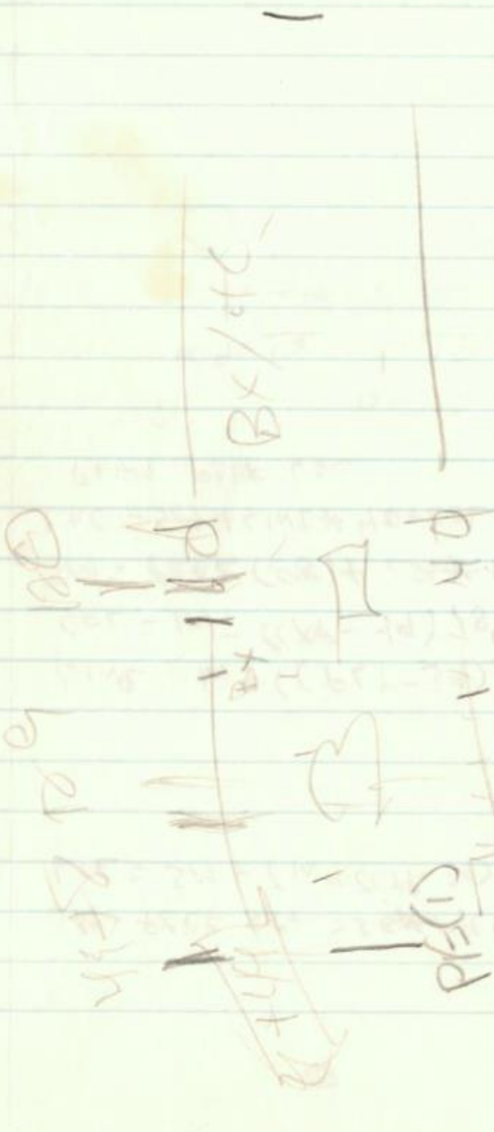
13

46

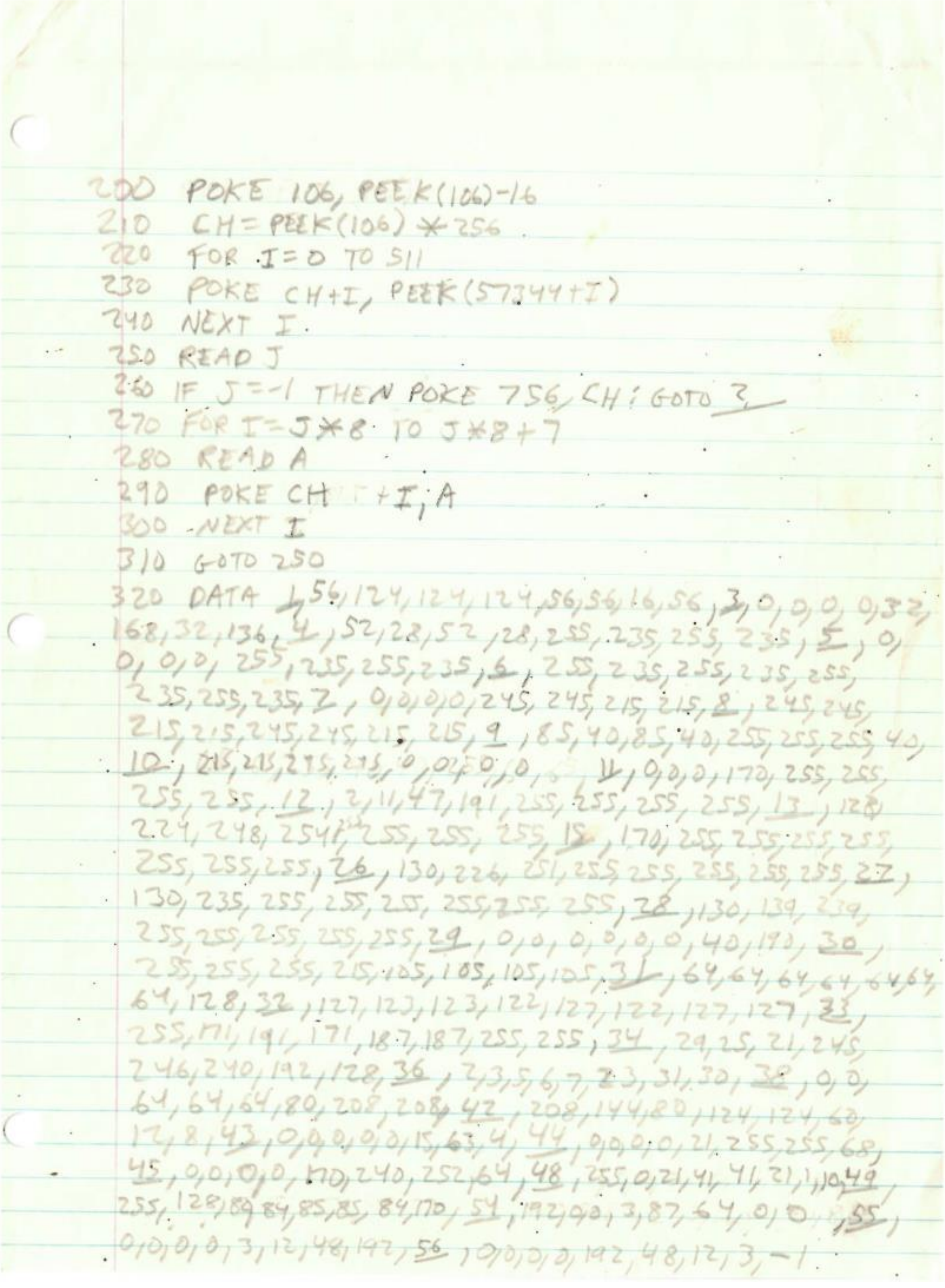
14
40

109 133 ~~BX-117~~

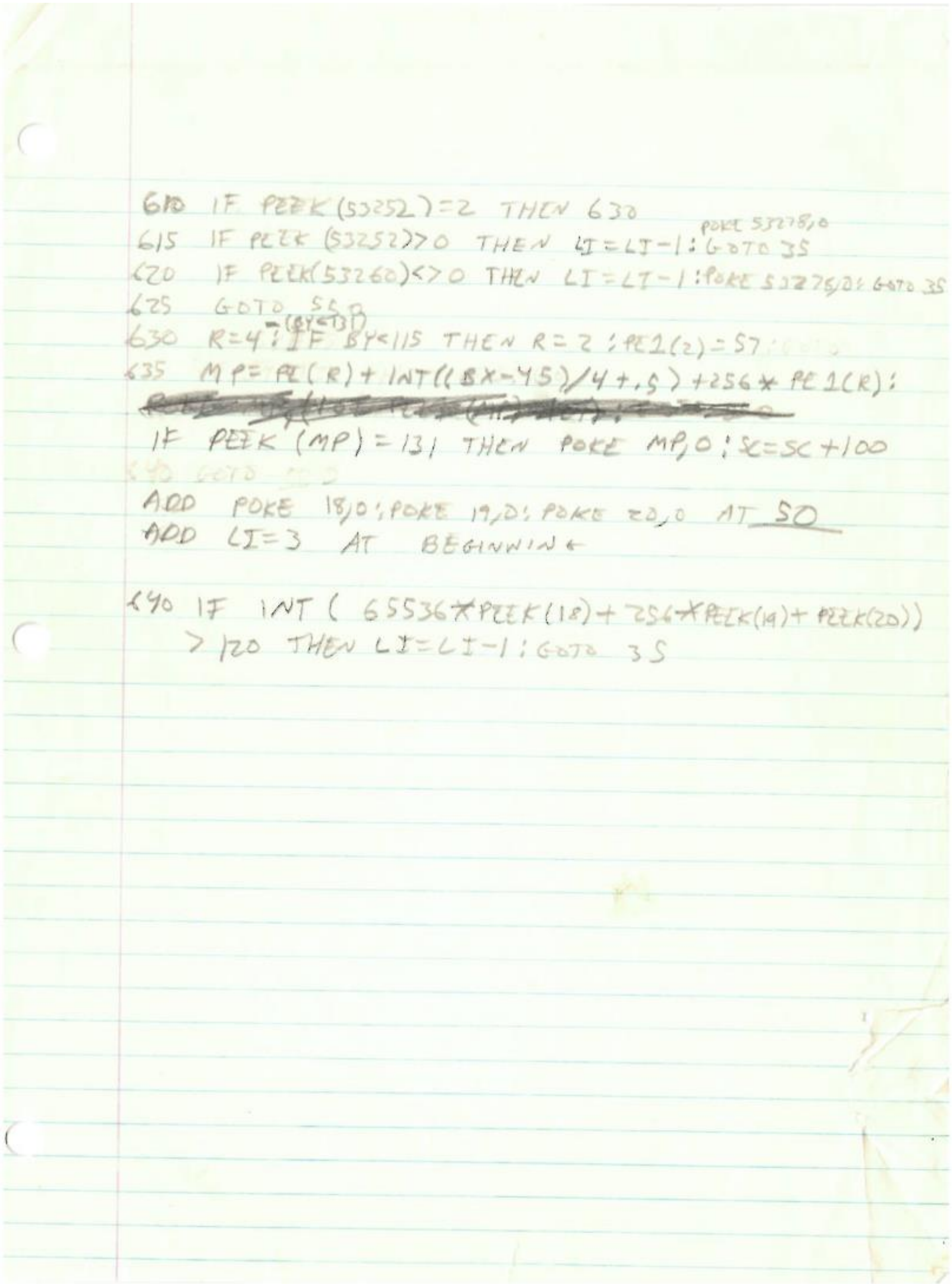
~~###~~ ABS(BY-105)



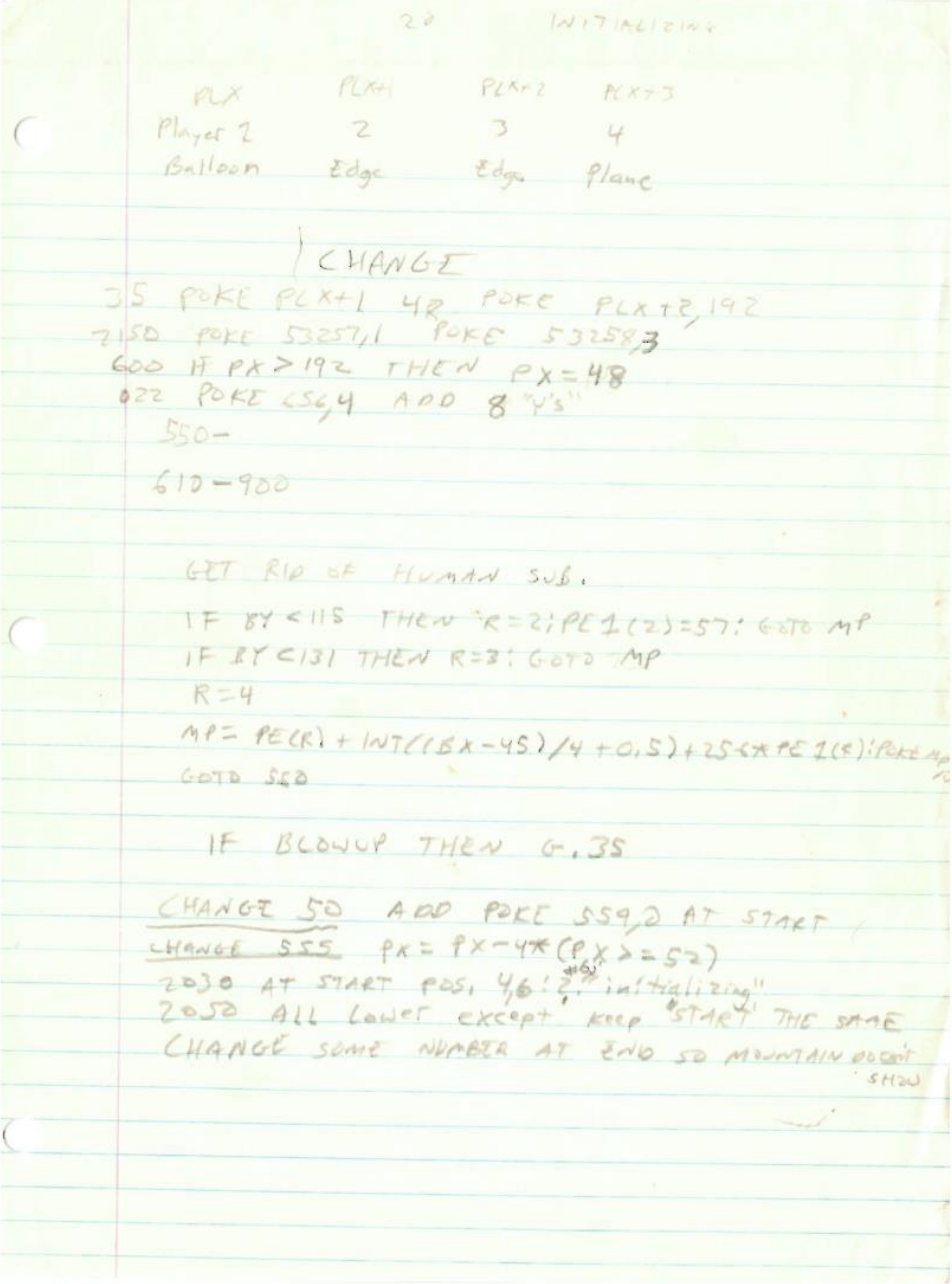
Bx/ctc
pd
4d
Pt=(1)
be
Bx



```
200 POKE 106, PEEK(106)-16
210 CH=PEEK(106)*256
220 FOR I=0 TO 511
230 POKE CH+I, PEEK(57344+I)
240 NEXT I
250 READ J
260 IF J=-1 THEN POKE 756,CH:GOTO ?
270 FOR I=J*8 TO J*8+7
280 READ A
290 POKE CH+I,A
300 NEXT I
310 GOTO 250
320 DATA 1,56,124,124,124,56,56,16,56,3,0,0,0,0,32,
168,32,136,4,52,28,52,28,255,235,255,235,5,0,
0,0,0,255,235,255,235,6,255,235,255,235,255,
235,255,235,7,0,0,0,0,245,245,215,215,8,245,245,
215,215,245,245,215,215,9,85,40,85,40,255,255,255,40,
10,215,215,245,245,0,0,0,0,11,0,0,0,170,255,255,
255,255,12,2,11,47,191,255,255,255,255,13,128,
224,248,254,255,255,255,15,170,255,255,255,255,
255,255,255,26,130,226,251,255,255,255,255,255,27,
130,235,255,255,255,255,255,255,28,130,139,239,
255,255,255,255,255,29,0,0,0,0,0,0,40,190,30,
255,255,255,215,105,105,105,105,31,64,64,64,64,64,64,
64,128,32,127,123,123,122,127,122,127,127,33,
255,171,191,171,187,187,255,255,34,29,25,21,245,
246,240,192,128,36,2,3,5,6,7,23,31,30,38,0,0,
64,64,64,80,208,208,42,208,144,80,124,124,60,
12,8,43,0,0,0,0,0,15,63,4,44,0,0,0,0,21,255,255,68,
45,0,0,0,0,170,240,252,64,48,255,0,21,41,41,21,1,10,49,
255,128,89,84,85,85,84,170,54,192,90,3,87,64,0,0,55,
0,0,0,0,3,12,48,192,56,0,0,0,0,192,48,12,3,-1
```



```
610 IF PEEK(53252)=2 THEN 630
                                           POKE 53278,0
615 IF PEEK (53252)>0 THEN LI=LI-1:GOTO 35
620 IF PEEK(53260)<>0 THEN LI=LI-1:POKE 53278,0:GOTO 35
625 GOTO 560
           -(BY<131)
630 R=4:IF BY<115 THEN R=2:PE1(2)=57
635 MP=PE(R)+INT((BX-45)/4+.5)+256*PE1(R):
    IF PEEK (MP)=131 THEN POKE MP,0:SC=SC+100

ADD POKE 18,0:POKE 19,0:POKE 20,0 AT 50
ADD LI=3 AT BEGINNING

640 IF INT( 65536*PEEK(18)+256*PEEK(19)+PEEK(20))
    >120 THEN LI=LI-1:GOTO 35
```



20 INITIALIZING

PLX	PLX+1	PLX+2	PLX+3
Player 2	2	3	4
Balloon	Edge	Edge	Plane

CHANGE

```
35 POKE PLX+1 48 POKE PLX+2,192
7150 POKE 53257,1 POKE 53258,3
600 IF PX>192 THEN PX=48
622 POKE 656,4 ADD 8 "Y's"
550-

610-900
```

GET RID OF HUMAN SUB.

```
IF BY<115 THEN R=2:PE1(2)=57: GOTO MP
IF BY<131 THEN R=3: GOTO MP
R=4
MP= PE(R) + INT((BX-45)/4+0.5)+256*PE1(R):POKE MP,
GOTO 560
```

IF BLOWUP THEN G.35

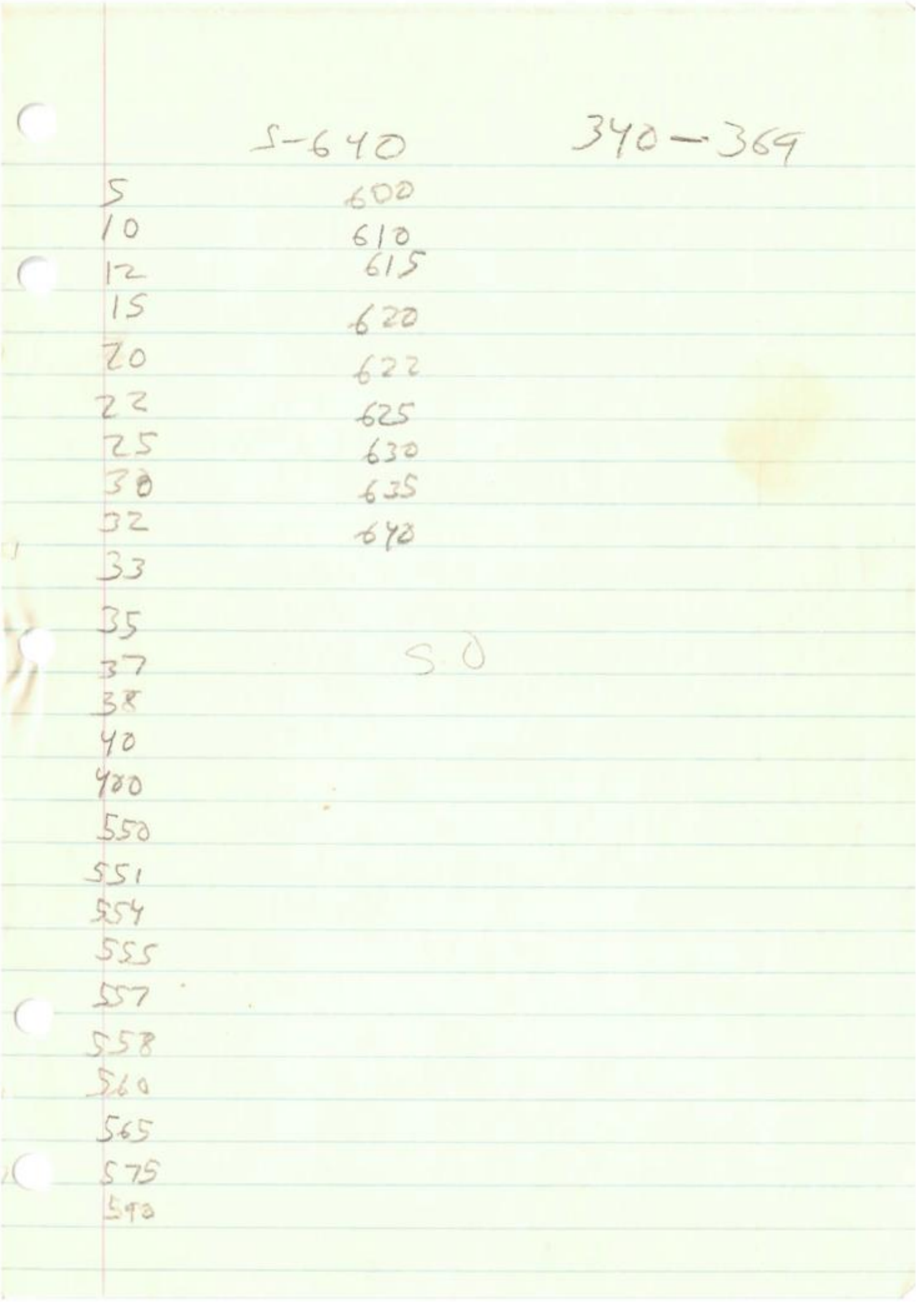
CHANGE 50 ADD POKE 559,0 AT START

CHANGE 555 PX= PX-4*(PX>=52)

2030 AT START POS. 4,6:?#6;"initializing"

2050 ALL Lower except keep "START" THE SAME

CHANGE some NUMBER AT END SO MOUNTAIN DOESN'T SHOW



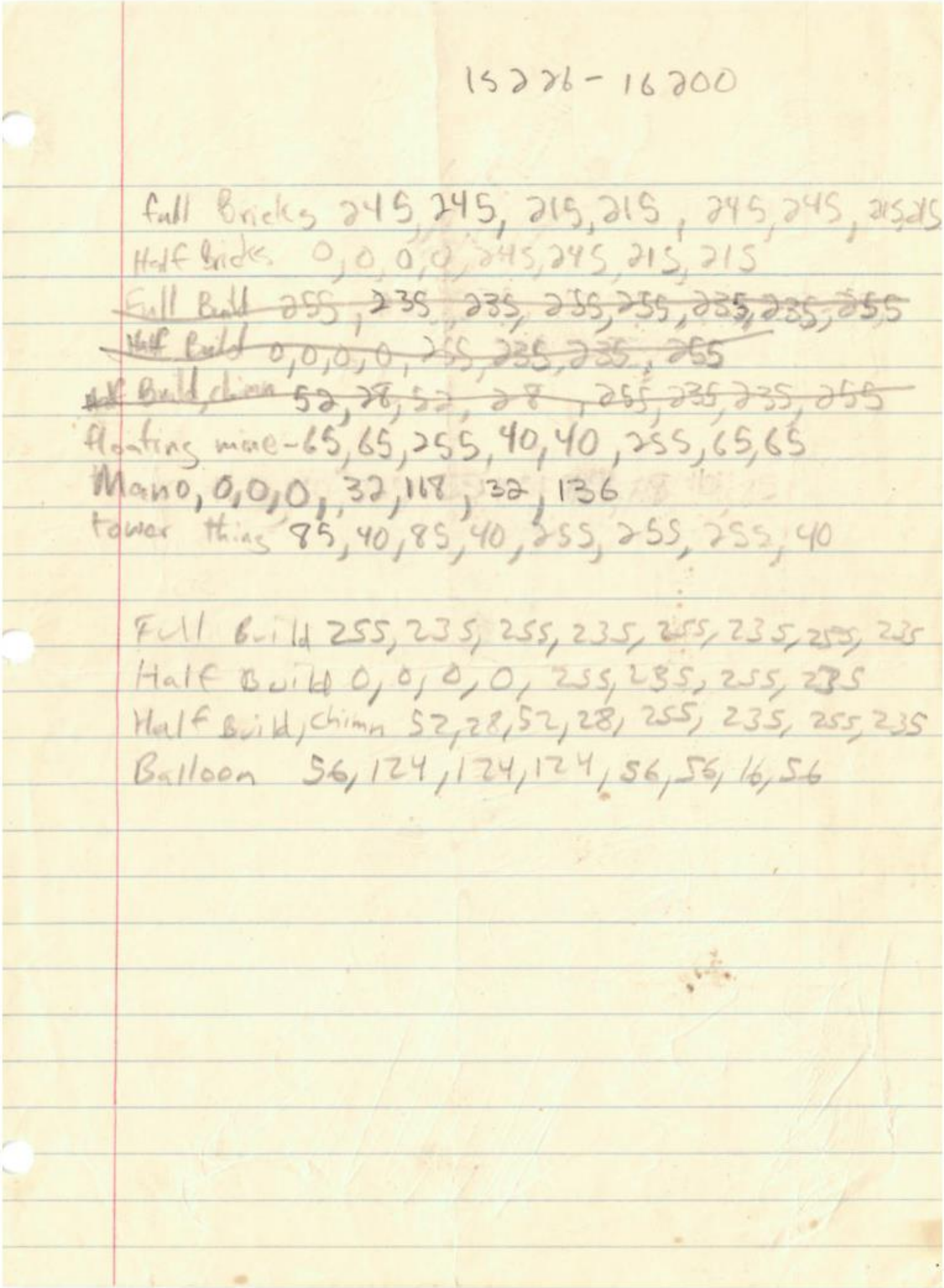
S-640

340—369

5
10
12
15
20
22
25
30
32
33
35
37
38
40
480
550
551
554
555
557
558
560
565
575
590

600
610
615
620
622
625
630
635
640

S.0



15226 - 16200

full Bricks 245, 245, 215, 215, 245, 245, 215, 215
Half Bricks 0, 0, 0, 0, 245, 245, 215, 215
~~Full Build 255, 235, 235, 255, 255, 235, 235, 255~~
~~Half Build 0, 0, 0, 0, 255, 235, 235, 255~~
~~Half Build, chimn 52, 28, 52, 28, 255, 235, 235, 255~~
floating mine - 65, 65, 255, 40, 40, 255, 65, 65
Mano, 0, 0, 0, 32, 118, 32, 136
tower thing 85, 40, 85, 40, 255, 255, 255, 40

Full Build 255, 235, 255, 235, 255, 235, 255, 235
Half Build 0, 0, 0, 0, 255, 235, 255, 235
Half Build, chimn 52, 28, 52, 28, 255, 235, 255, 235
Balloon 56, 124, 124, 124, 56, 56, 16, 56

15984

	128	64	32	16	8	4	2	1	
SE.0		X		X		X		X	GREEN
SE 1	X		X		X		X		YEllow

SE. 2 is Both RED

SE. 0, $\overset{12}{X}$, 6 — GREEN = 198
SE. 1, $\overset{13}{X}$, 10 - YEllow = 218
SE. 2, $\overset{3}{X}$, 2 - RED = 50
SE. 4, $\overset{0}{X}$, 0 - Blue = 0

Mode 1 switched Background and Balloon

Balloon is Black

contact with yellow alone gives bonus points all other combinations causes blow-up

Add High Score feature
Start where you left off when you get blown up
add landing spot at end of set
Poke 77, 0
Redefine Nazi Flag
— Add sound for landing on platform
Add a clause in if (landing on platform)
that makes BY > height of helicopters
Add a continual sound!

Redefine German Flag
Change bonus at end so
time will not be a major
factor if not blown up before
end. (Keep track of time for whole set.
Do not reset time when blown up)

Helicopter
Visible at left

Poke 656, 1 2

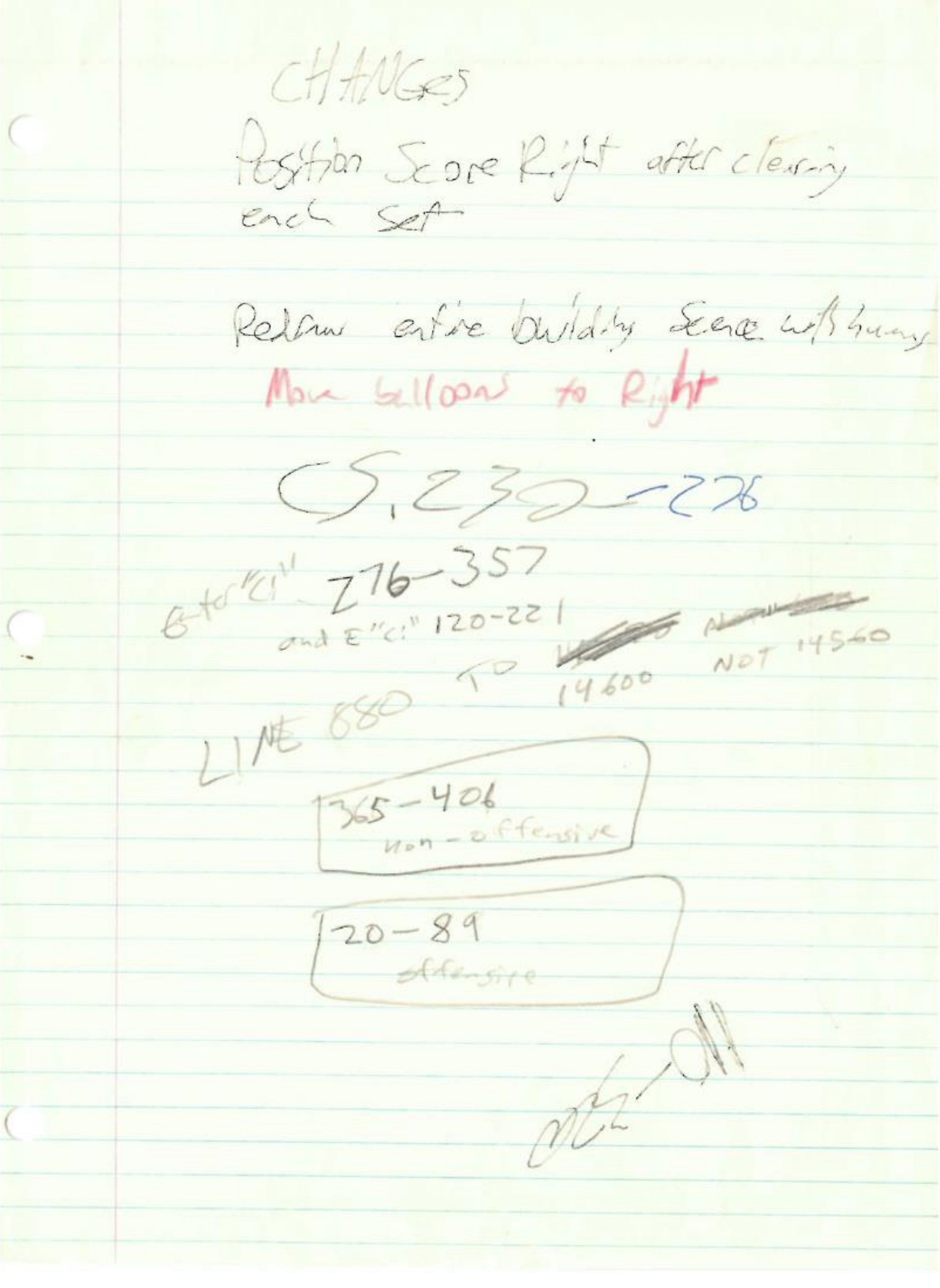
Poke 657, 1 2

?, "Hi score";HS

HI SCORE 9 1000

SO, 0, 180, 2, 2 ;3
SO, 1, 181, 0, 1
SO, 2, 182, 2, 3

```
1080 FOR EA = 14336 TO 14535 POKE EA,32
 NEXT EA ,POKE 550,0

1090 FOR EA = 14536 TO 14802 POKE
  EA,0 NEXT EA GOTO 150

1060 NEXT JW POKE 256,CM/256 GOTO 150
```



CHANGES

Position Score Right after clearing each set

Redraw entire building Scene with humans

Move balloons to Right

CS,230-226

Enter "C:" 276-357

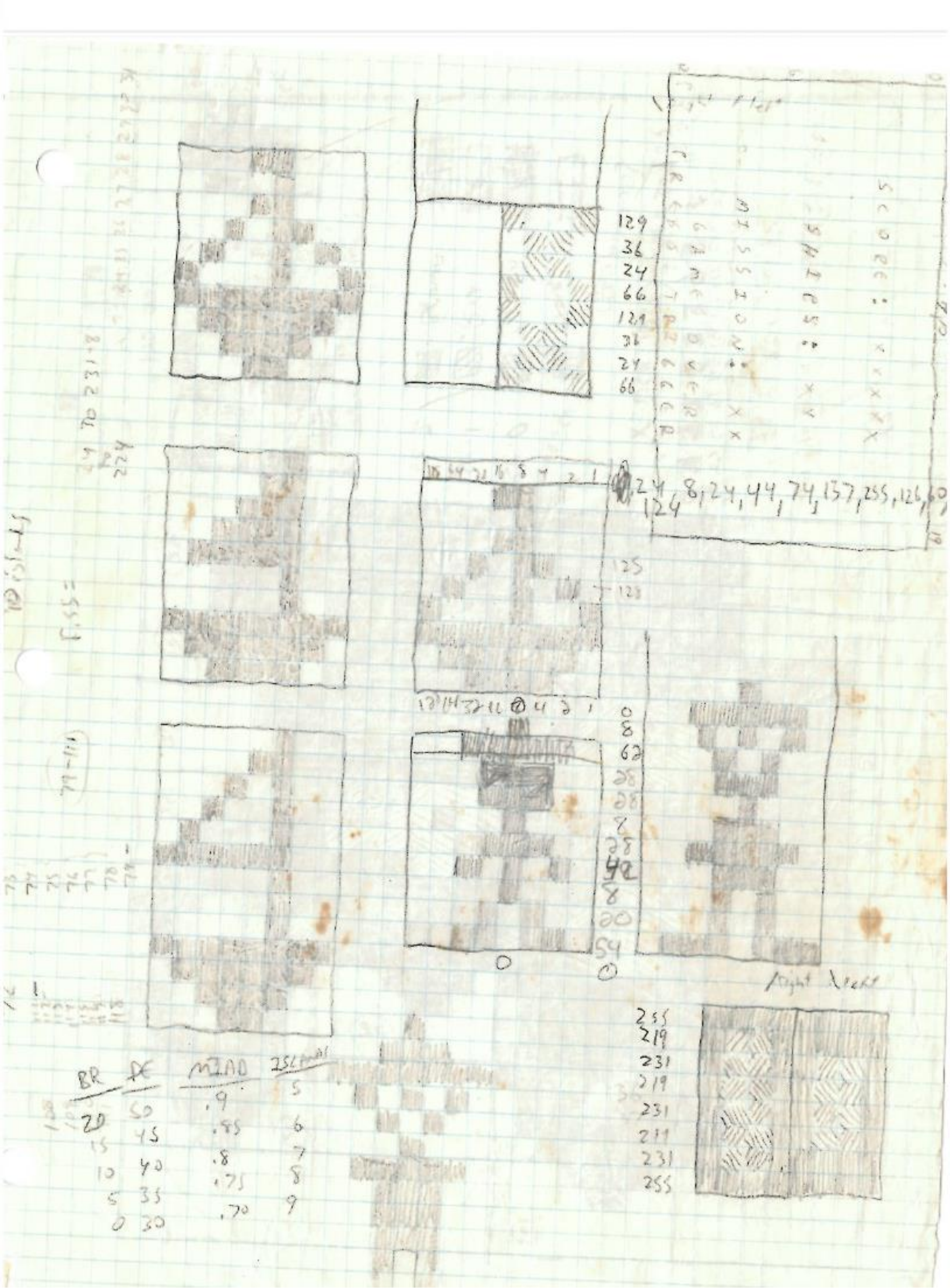
and E"C:" 120-221

LINE 880 TO 14600 NOT 14560

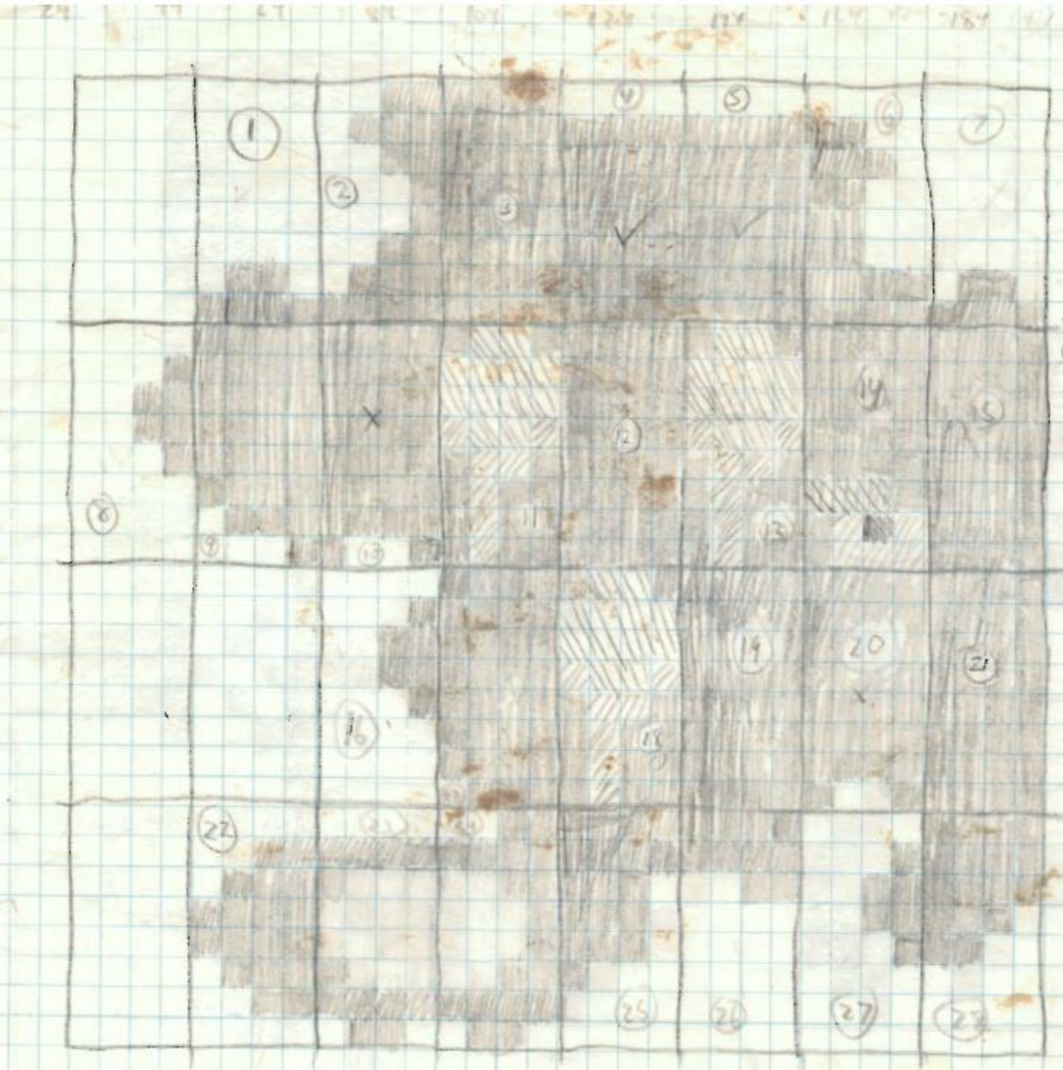
365-406
non-offensive

20-89
offensive

PIRACY II



BR	DE	MIAD	ISLANDS
20	50	.9	5
15	45	.85	6
10	40	.8	7
5	35	.75	8
0	30	.70	9



RND (72-118) *256 + Horizontal (0-250)

24 46 68 90 112 134 156 178 200 222

25 TO 224 , STEP 22

25 47 69 91 113 135 157 179 201 223

0 4 7 2 6 8 3 9 5 1

X=USR(1536): POKE 559,62

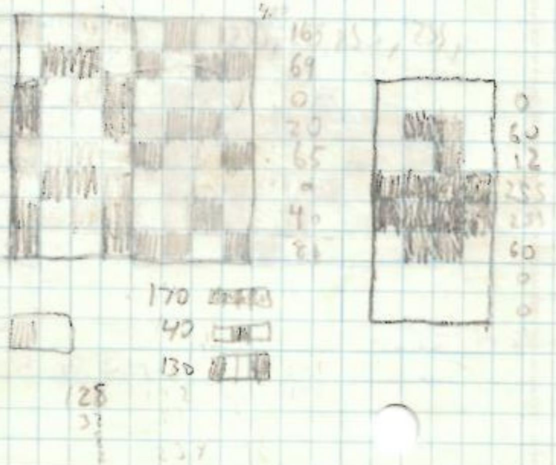
(BACKGROUND)

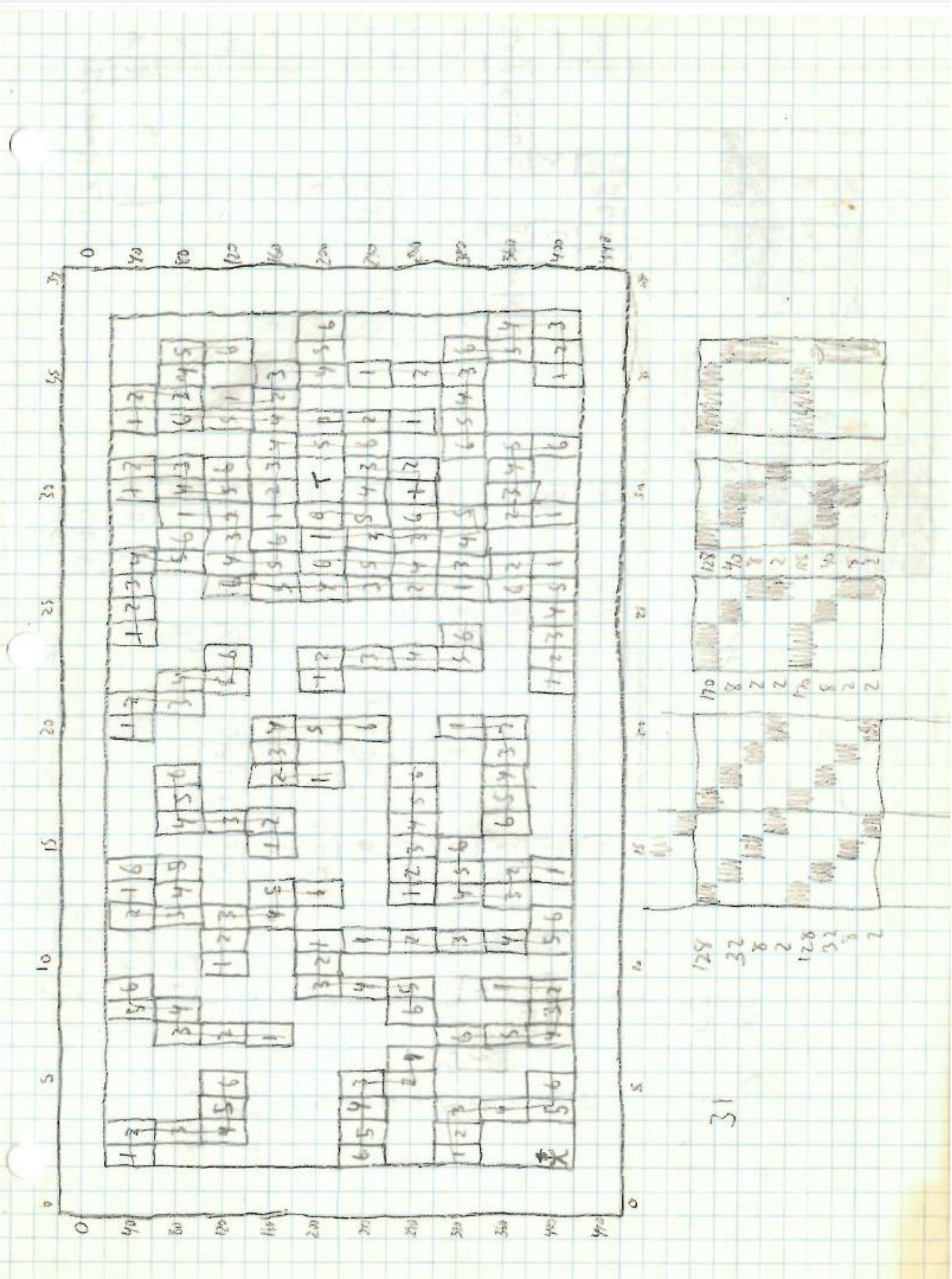
Se.4 ▥ Blue ▭▭

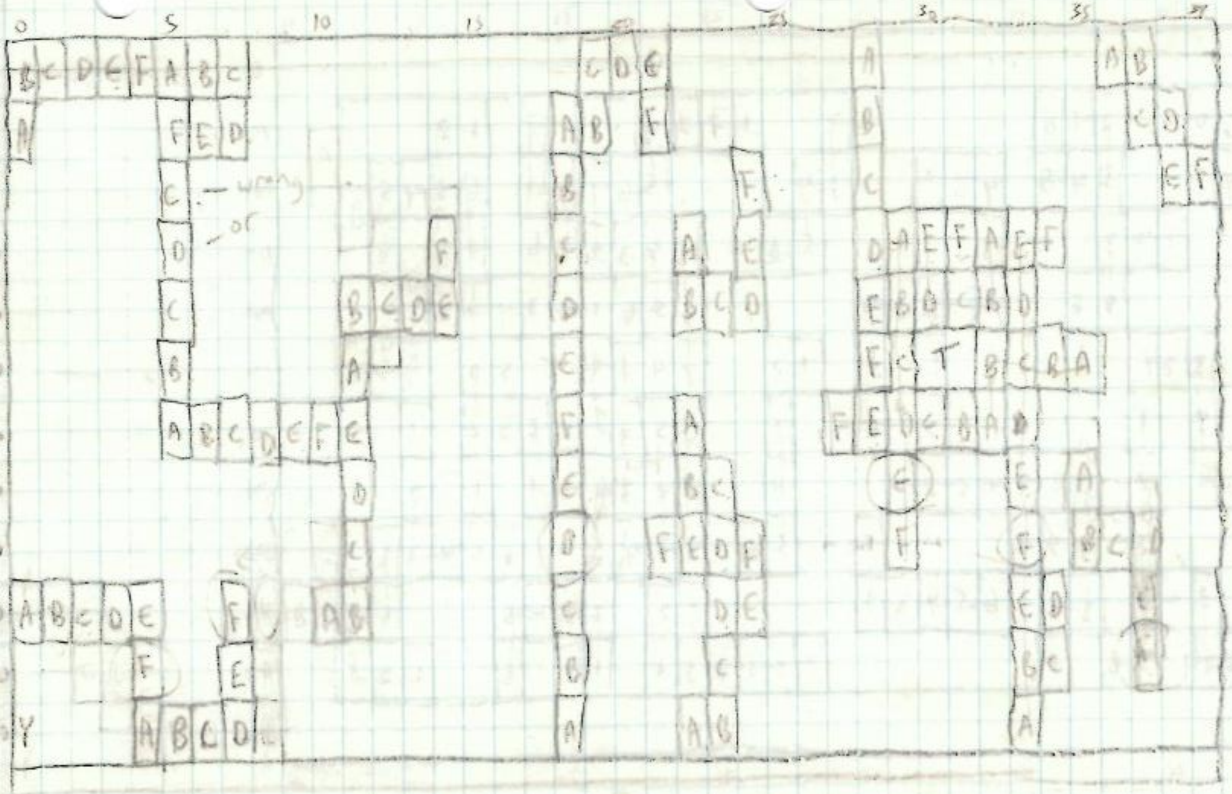
Se.0 ▧ Green ▭▥

Se.1 ▨ Red ▥▭

Se.2 ■ Brown ▥▥




0
40
80
120
160
200
240
280
320
360
400
440
0
5
10
15
20
25
30
35
37
T
X
128
32
8
2
170
31



19 22 21

22 21 19

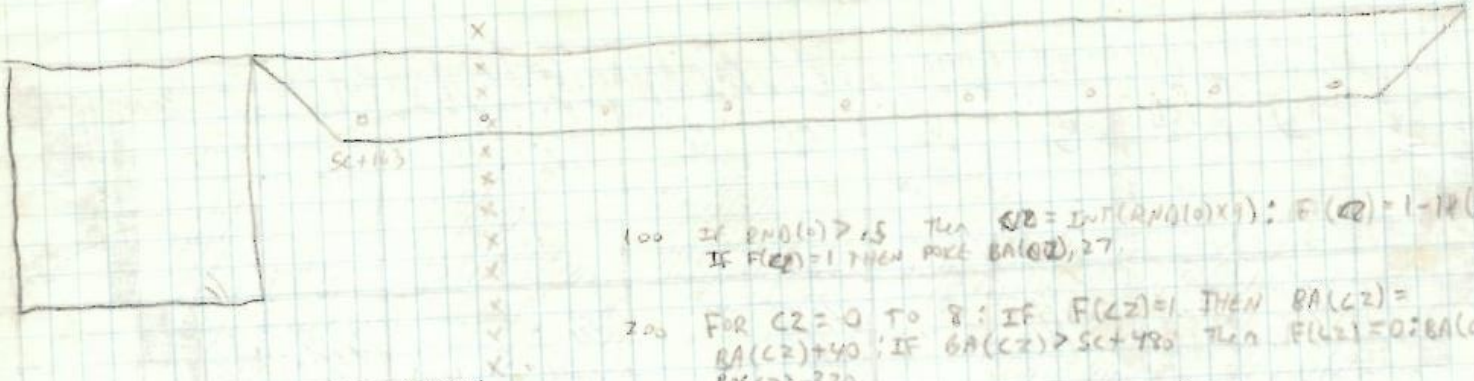
0
1
2
3
4
5
6
C-7
8
C+1 9
10
11

left: 0,0,85,21,15,15,15,15

right: 0,0,80,84,255,175,255,255.

20 23 23

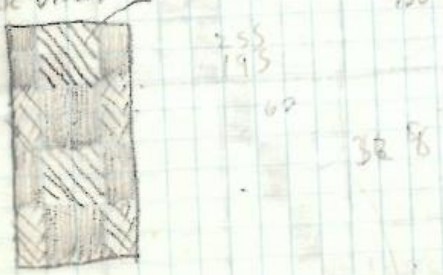
22 22 19



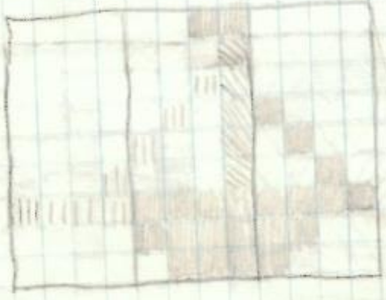
```
100  IF RND(0)>.5 THEN QR=INT(RND(0)X9): F(QR)=1-1*(F(QR)=1):
     IF F(QR)=1 THEN POKE BA(QR),27

200  FOR CZ=0 TO 8: IF F(CZ)=1 THEN BA(CZ)=
     BA(CZ)+40 :IF BA(CZ)>SC+480 THEN F(CZ)=0:BA(CZ)=
     BA(CZ)-320
300 IF F(CZ)=1 THEN POKE BA(CZ)-40,0: POKE BA(CZ),27
400 N.CZ :GOTO 100
```

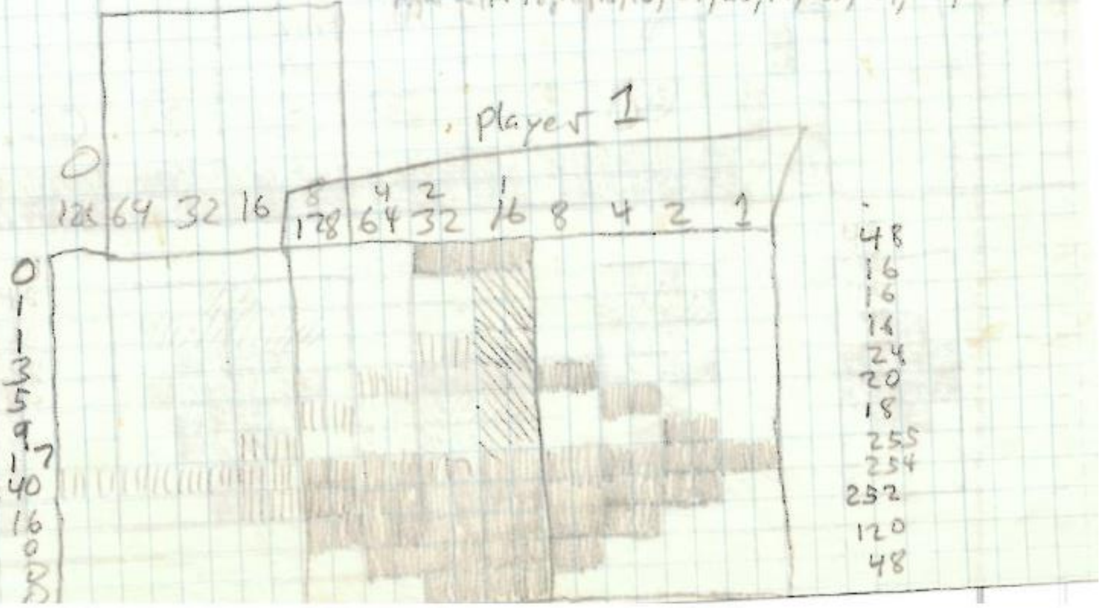
BORDER: 235,235,125,125,235,235,125,125

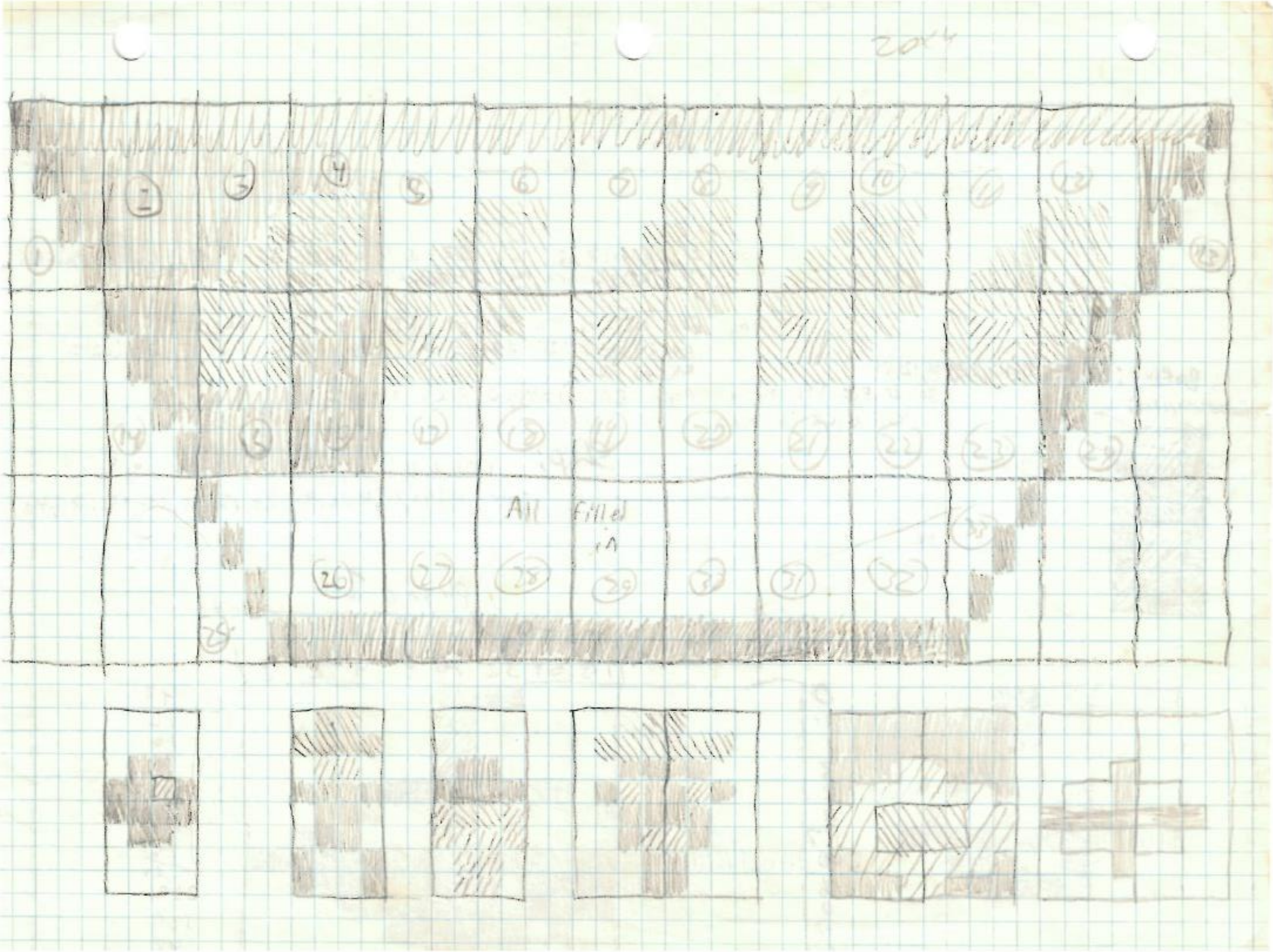
(Reverse Video)



Ship: left half: 0,1,1,3,5,9,17,240,16,0,0,0
right half: 48,16,16,16,24,20,18,255,254,252,120,40






20x4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
All filled in

Whirlpools in the water.

Start with 5 islands

and same number of ships.

All ships start with a direction table (1, -1, 256, -256) to determine where they go. If they run into an island then make them take the treasure, and poke it away so it doesn't show on the island (same as if you took it). When all 5 treasures are gone, increase number of islands (max of 15 or so) and decrement machine language delay for firing cannons (get a random gun and shoot ship), Islands have to be at a set position horizontally (as they are now) to determine which one you hit,

rocks on border and land beyond (nongreen character)

As many Pirate ships on screen as islands. Must first touch pirate ship then go to battle scene after this you have to land on island.

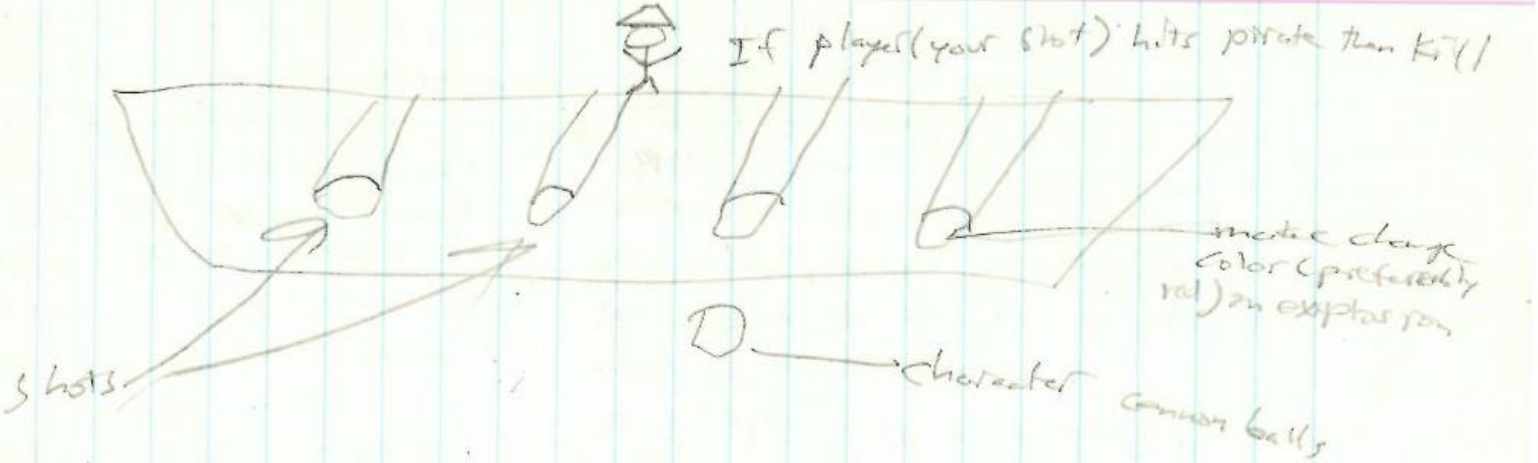
If you don't battle ship and touch island, you die because your ship isn't big enough to land on the island (we'll make you double width and maybe change your death once you win battle).

You can't battle two pirate ships in a row (go right through them) because the other ship thinks you are a pirate ship and won't fight,

Islands originally have treasure on them and when you get through island and touch screen, it disappears from island (have to use arrays to tell where islands are)

Have to make pirate ships on screen in reverse video but distinguish from rocks by making some color intermixed in rocks

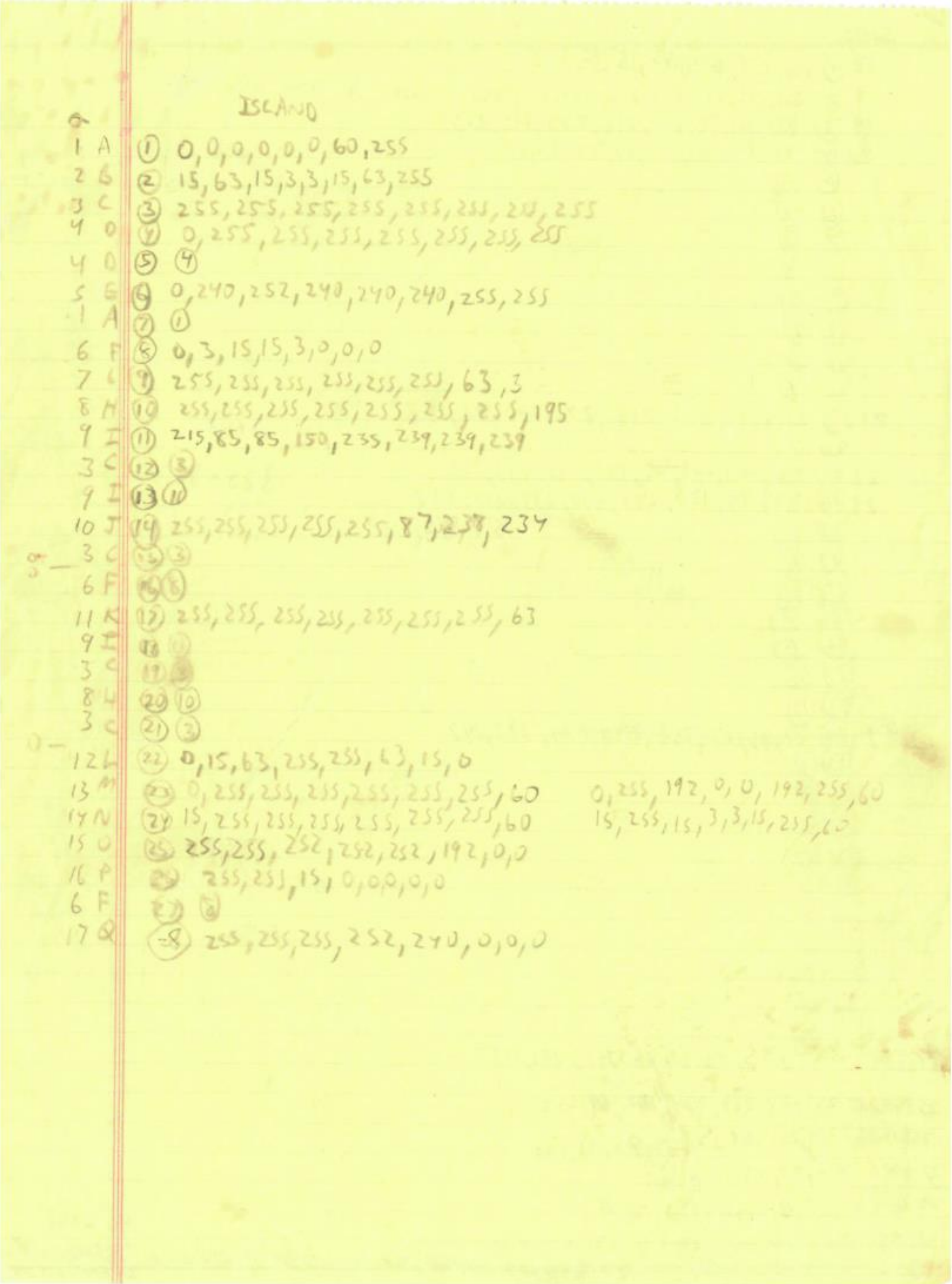
If player (your shot) hits pirate than Kill



You have a raft
that is too small to

make it into turbulent waters near island,
So you must take over the pirate
Ship in order to land on island,
And you must get map from pirate.
Once take over a ship, can land
at any Island

YOU if player (your ship) hits
any playfield Then Kill



ISLAND

o

1 A ① 0,0,0,0,0,0,60,255
2 B ② 15,63,15,3,3,15,63,255
3 C ③ 255,255,255,255,255,255,255,255
4 D ④ 0,255,255,255,255,255,255,255
4 D ⑤ ④
5 E ⑥ 0,240,252,240,240,240,255,255
1 A ⑦ ①
6 F ⑧ 0,3,15,15,3,0,0,0
7 G ⑨ 255,255,255,255,255,255,63,3
8 H ⑩ 255,255,255,255,255,255,255,195
9 I ⑪ 215,85,85,150,235,239,239,239
3 C ⑫ ③
9 I ⑬ ⑪
10 J ⑭ 255,255,255,255,255,87,238,234
o — 3 C ⑮ ③
6 F ⑯ ⑧
11 K ⑰ 255,255,255,255,255,255,255,63
9 I ⑱ ⑪
3 C ⑲ ③
8 H ⑳ ⑩
3 C ㉑ ③
0 — 12 L ㉒ 0,15,63,255,255,63,15,0
13 M ㉓ 0,255,255,255,255,255,255,60 0,255,192,0,0,192,255,60
14 N ㉔ 15,255,255,255,255,255,255,60 15,255,15,3,3,15,255,60
15 O ㉕ 255,255,252,252,252,192,0,0
16 P ㉖ 255,255,15,0,0,0,0,0
6 F ㉗ ⑧
17 Q ㉘ 255,255,255,252,240,0,0,0

SHIP:

18 1 255, 255, 63, 63, 15, 15, 3, 3
3 2 255, 255, 255, 255, 255, 255, 255, 255   use 3 from island
19 3 255, 255, 255, 255, 255, 253, 245, 213
20 4 255, 255, 255, 255, 87, 87, 87, 87
5 3
6 4
7 3
8 4
9 3
10 4
11 3
12 4
21 13 255, 255, 252, 252, 240, 240, 192, 192
14 1
22 15 85, 105, 105, 85, 255, 255, 255, 255
23 16 87, 95, 127, 255, 255, 255, 255, 255
17 15
18 16
19 15
20 16
21 15
22 16
23 15
~~24 24 87, 95, 124, 252, 240, 240, 192, 192~~
25 1
26 2
27 2
28 2
29 2
30 2
31 2
32 2
33 13

25 characters

WAVE

320-327

o

200

9 PIRATE: 20, 85, 40, 255, 60, 60, 195, 195

25 PortholeA: 255, 255, 254, 234, 169, 169, 234, 254

26 PortholeB: 255, 255, 191, 171, 90, 90, 171, 141

37 BALL: 0, 0, 20, 89, 85, 20, 0, 0

28 Left trees: 0, 8, 35, 21, 13, 15, 15, 11

29 Right trees: 0, 0, 82, 34, 255, 175, 255, 251

30 – 35 – Rocks on isld   27 BORDER CHARACTER 36 LAND ON BORDER

40 WAVE
38 border on 3rd screen
39 Enemy ships

M=PEEK(106)-64 : N=M*256

B1
B2
J1 T1
J2 T2
J3
J4 T3
T4
B3
E3
E5
T5
E2
E1

1 2 3 4 dift

SYNMAIN, SYNCASS, ROCKYOU, BRICKLAY,

MINUETGM BRAHMS VICTORS

Pokey 5 Pokey 6 7 Pokey

Minuet Brahms VICTORS Nowhere Man 10-32

0-20 32-60 70-102 Under Pressure 50-90

SYNMAIN SYNCASS RockYou Bricklay

110-152 160-172 180-192 200-235

SYNMAIN - machine

250-291

31A

768 26

794

[10-74 Am42

[100-134 Creator

[150-188 maze

[200- game

Heim ~~Py Pog~~ ~~Popeye~~ ✓Beta Lyrae ~~Zaxxon~~

~~Jungle Hunt~~ Computer War

Weis ✓Popeye ✓Ralley Speedway ✓Tails of Beta Lyrae

line dl D=PEEK (106)-14
50 ND = DX256
12800
12864 DL+64
12875 DL+75
M 13056 DL+256
13295 DL+495
14380 DL+1580
15107 DL+2307
420 D8=D+8 R=56 R=D+6
M1 13311 DL+511
13255 DL+455
13256 DL+456

427-1402
538-0197
544-0885
582-0657
978-8087
Lidi 553-4005
882-5409
559-1676
459-4531
Software House 776-9742
771-4126
368-4828

DL+7, MEM-14 D
DL+12, MEM-13 D-1
DL+16, MEM-8 D-6
DL+21, MEM-7 D-7
DL+24, MEM-6 D-8

PM=PEEK(106)-16

PHA 40
EDA 30
S: STA 53248
ADC #32
BNE S
PLA
RTI

48

72
24
216
169,0
141,0,208
105,32
208, 249
104 64

ARCADE BBS
37220 TRICIA DRIVE
STERLING HEIGHTS, MI 48007
$ISTAPE

A

TELEX

147 58
190-[illegible]

M

✓AMODEM42
✓AMY200C
~33600/3722 SH
~~SA~~
SA AND

PLAYERS ~~[illegible]~~

~~DOUBLE WIDTH~~ 11 → 7

1775=CAN 1771- flag to stop

FLAG=1776

DATA 1777–1780 0,3,3,0

1782-YX 1781-SY

whatever to wherever +133 (including VB setup)

```
        CLD             216
        LDA 632         173,120,2
        CMP #11         201,11
        BNE RIGHT       208,16
        SEC             56
        LDA YX          173,246,6
        SBC #1          233,1
        CMP #50         201,50
        BCC CONT1       144,3
        STA YX          141,246,6
CONT1   CLC             24
        BCC DONE        144,17
RIGHT   CMP #7          201,7
        BNE DONE        208,13
        CLC             24
        LDA YX          173,246,6
        ADC #1          105,1
        CMP #200        201,200
to fix  BCS DONE        176,3
        STA YX          141,246,6
DONE    LDA YX          173,246,6      ADC #7
        STA 53248       141,0,208      STA 53249
105,7
141,1,208
        LDA CAN         173,239,6      CAN = 1 shot in air
        BEQ CONT2       240,30             0 can shoot
        LDX #0          162,0          LDY SY
        LDY SY          172,245,6      LDA #0
LOOP    LDA DATA,X      189,241,6      STA (MS),Y
        STA (PM),Y      145,207        INY
        INX             232            STA (MS),Y
        INY             200            DEY
        CPX #4          224,4          DEY
        BNE LOOP        208,245        LDA #3
        DEC SY          206,245,6      STA (MS),Y
        LDA SY          173,245,6      DEY
        BNE CONT3       208,27         STA (MS),Y
        STA CAN         141,239,6
        CLC             24
```

```
CONT2   LDA STRIG0 644          173,132,2
        BNE CONT3               208,16
        LDA #200                169,200
        STA SY                  141,245,6
        LDA YX                  173,246,6
        ADC #                   105,13
        STA MISHP 53252         141,4,208
        STA CAN                 141,239,6
CONT3   LDA MISSLE to playfield 53248   173,0,208
        AND #8                  41,8
        BEQ CONT4               240,5
        LDA #1                  169,1
        STA FLAG                141,240,6
CONT4   JMP XITVB or scrolling  76,98,228
```

Variables

yourx shoty canshot 1 if shot in air 0 if can shoot 1 if hit pirate 0 if not

YX, SY, CAN, FLAG

DATA 0,3,3,0

X=USR(1536,512)

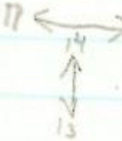
→ 2ρ in memory not ~~02~~

```
whatever to                         207,203                30
whatever +44
                                                           240
       PLA                  104
       PLA                  104
       PLA                  104        POKE 207,240
       STA  203             133,203    POKE 208, CH
       LDY  203             164,203
       LDX  #0              162,0
Loop   LDA  DATA,X          189,247,6
       STA  (207),Y         145,207                DATA
       INY                  200
       INX                  232                     20
       CPX #8               224,8                   85
       BNE  LOOP            208,245                 40
       LDA #0               169,0                   255
       STA 20               133,20                  60
LOOP2  LDA 20               165,20                  60
       CMP #30  CMP 1791    ~~201,30~~ 205,255,6    195
       BNE LOOP2            208, 249  249           195
       LDY  203             164,203                      poke it
       LPA  #0              169,0              1783      here
       TAX                  170                1790
LOOP3- STA (207),Y          145,207
       INY                  200
       INX                  232
       CPX #8               224,8
       BNE LOOP3            208,248
       RTS                  96
```

```
GR.0
DL=PEEK(560)+256*(PEEK(561)+4
POKE DL-1,69
FOR I=2 TO 12:POKE DL+I,5:NEXT I
POKE DL+13,65
```

112 0C+0
112 1
112 2
71 3 4
112
138 5
7 6
7 14
66 15
96 16
159 17
2
2
2
65
88
155

12 Pages: Peek(106) – 16 Ant/Ch set –14 2 pages
– 12 players
– 8 6 pages
–2 –DL and GR. 2
–20 4 pages

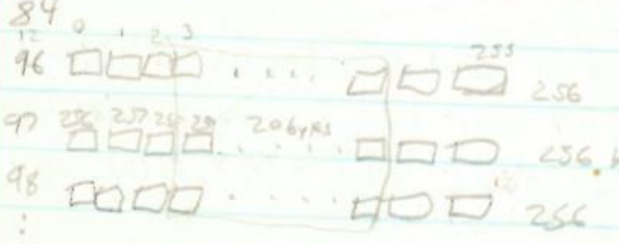


Poke all LMS's same number

10 0
10 256

72
84

Page



256 –256 up
+256 down
256 bytes –1 left
256 +1 right

3K
:
9K 107
12K 119

12X4 = 48 pages gives a 6X4 screen
12k for 6X4

```
DIM PE(11),PE1(11)
```

~~DL=50*256:FOR~~

```
10 FOR X=0 TO 11
20 PE(X)=PEEK(DL+4+3*X):PE1(X)=PEEK
   (DL+5+3*X):N.X
30 IF STICK(0)=14 THEN FOR X=0 TO 11:PE1(X)=
   PE1(X)-1:NEXT X
40 IF STICK(0)=13 THEN FOR X=0 TO 11:PE1(X)=
   PE1(X)+1:N.X
50 IF STICK(0)=11 THEN FOR X=0 TO 11:PE(X)=PE(X)-1:N.X
60 IF STICK(0)=7 THEN FOR X=0 TO 11:PE(X)=PE(X)+1:N.X
70 FOR X=0 TO 11:POKE DL+4+3*X,PE(X):POKE
   DL+5+3*X,PE1(X):N.X
80 G.30
```

632=STICK(0)

DL+16 54276=hor

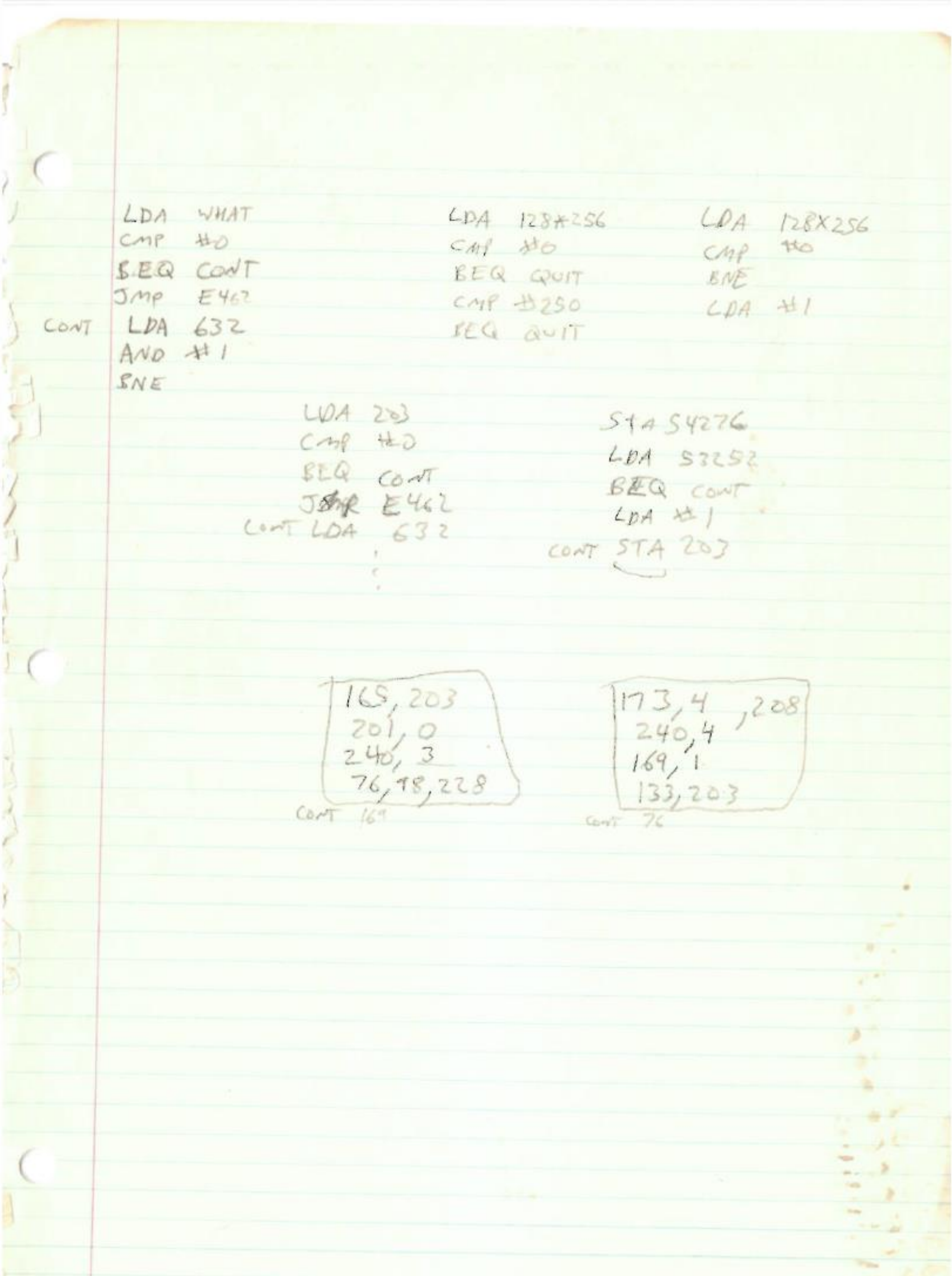
DL+32 54277=vert

STKADD 0,0,0,0,0,1,1,1,0,255,255,255,0,0,0,0

G.C. 0-65

Shows 25-175

Vanguard 185-



```
     LDA  WHAT
     CMP  #0
     BEQ  CONT
     JMP  E462
CONT LDA  632
     AND  #1
     BNE
```

```
LDA  128*256
CMP  #0
BEQ  QUIT
CMP  #250
BEQ  QUIT
```

```
LDA  128X256
CMP  #0
BNE
LDA  #1
```

```
     LDA  203
     CMP  #0
     BEQ  CONT
     JSR  E462
CONT LDA  632
```

```
     STA 54276
     LDA 53252
     BEQ CONT
     LDA #1
CONT STA 203
```

```
165, 203
201, 0
240, 3
76, 98, 228
CONT 169
```

```
173, 4 , 208
240, 4
169, 1
133, 203
CONT 76
```

```
                                                       10 (0|0|0|0|1|0|1|0)
                                                       20 (0 0 0 1 0 1 0 0)
      CLD                                   165,203
      LDA  203                              201,60
      CMP  #60                              208, 7
      BNE  START
      LDA  #0                               169,0
      STA  203 — CLC BCC BYE
START LDA  632                              133,203          559
      CMP  #11                              24               512
      BNE  BYE                              144,18           ---
      LDX  #0                        START  173,120,2         47
LOOP  DEC  128+256,X                        201,11
      INX                                   208,11
      TXA                                   162,0             14
      CMP #12                        LOOP   222,0,128         16
      BCC LOOP                              232               84
BYE:  LDX #0      X  A  Y  206              138               46
LOOP  TXA         10 10 34                  201,12           ---
      ASL A   X2     20     10              144,247          224
      STX 206        30                                      229
                     34
      CLC                            BYE    162,0
      ADC 206                        LOOP   138   TXA
      ADC #4  — STA 206                     10    ASL
      TAY         133,206                   134,206
      LDA FE,X                              24
      STA (207),Y          INC 206          101,206
      INY  ———  LDY 206   164,206           105,4       58463
      LDA FE1,X                             168         95, 228
      STA (207),Y                           189,0,128
      INX                                   145,207
      TXA                                   200
      CMP #12                               189,0,129
      BCCLOOPY — L                          145,207
      JMP E462                              232
                                            138
                                            201,12         ~~LDA #0 169,0~~
                                            144,229 227    ~~STA 559 141,47,2~~
                                            76,98,228
```

150, 1030
2"!&↓←←+*↓←←+*↓←←
#@ ((()

```
                                            166,206
      LDA 632        173,120,2
      CMP #15        201, 15                    16
      BNE —          208, 3                      8
      76,98,228      76,98,228                 128
                                                14
                                               142
```

21

5
16
9
144
x
152

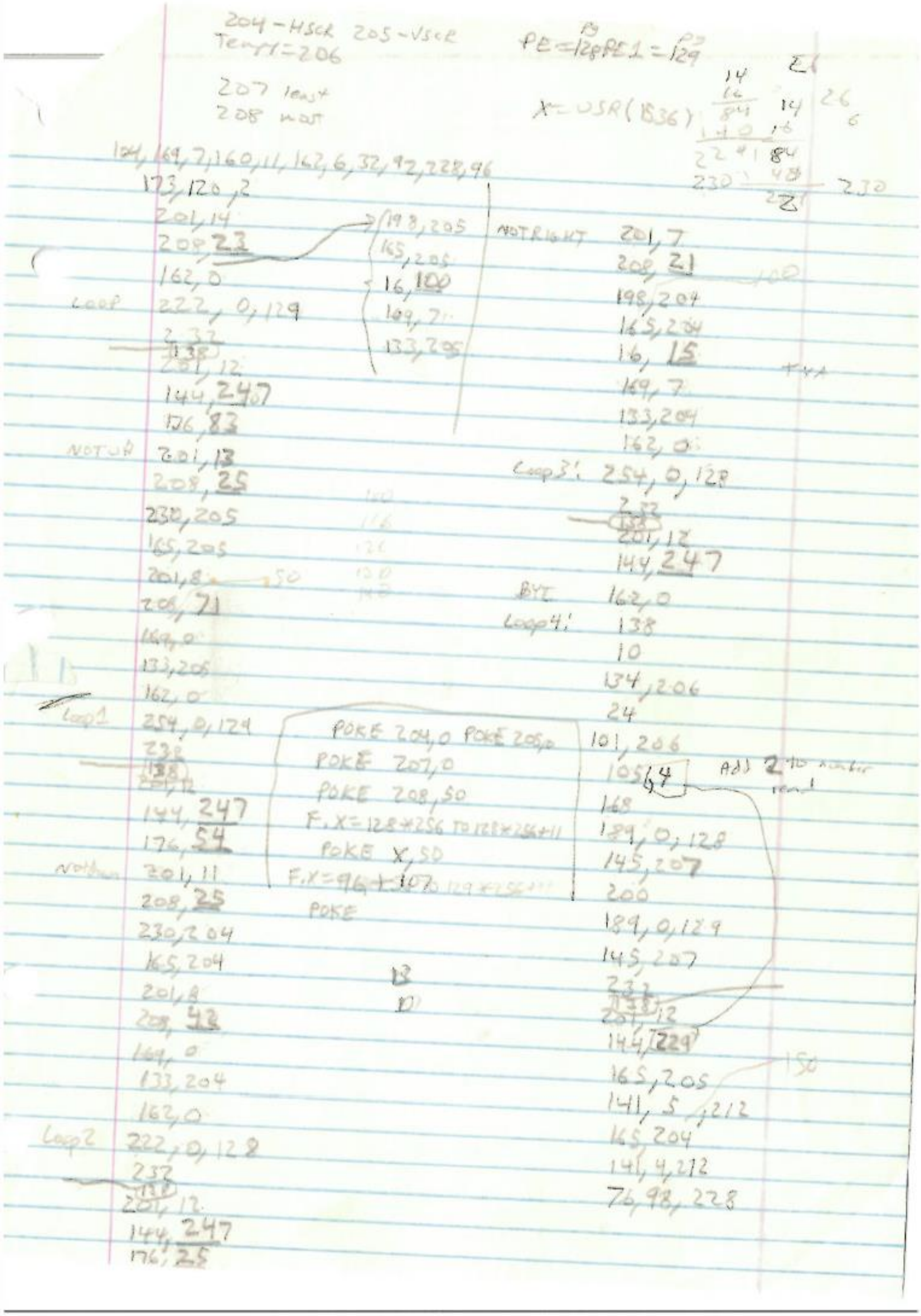
PHA	72
TXA	138
PHA	72
TYA	152
PHA	72

43 bytes

141,10,212

-29

PLA	104
TAY	166
PLA	104
TAX	170
PLA	104
RTI	64



204 - HSCR 205 - VSCR
TEMPY=206

207 least
208 most

PE=128 PE1=129

X=USR(1536)

14
16
84
140
224
230

14
16
84
48
230

26
6

```
104,169,7,160,11,162,6,32,92,228,96
      173,120,2
      201,14
      208,23
      162,0
LOOP  222,0,129
      232
      ~~138~~
      ~~201,12~~
      144,247
      176,83
NOTUP 201,13
      208,25
      230,205
      165,205
      201,8   50
      208,71
      169,0
      133,205
      162,0
Loop1 254,0,129
      232
      ~~138~~
      ~~201,12~~
      144,247
      176,54
notdown 201,11
      208,25
      230,204
      165,204
      201,8
      208,43
      169,0
      133,204
      162,0
Loop2 222,0,128
      232
      ~~138~~
      201,12
      144,247
      176,25
```

```
198,205
165,205
16,100
169,7
133,205
```

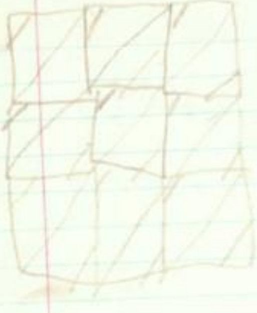
```
NOTRIGHT 201,7
         208,21   100
         198,204
         165,204
         16,15     TYA
         169,7
         133,204
         162,0
Loop3:   254,0,128
         232
         ~~138~~
         201,12
         144,247
BYT      162,0
Loop4:   138
         10
         134,206
         24
         101,206
         105,64   Add 2 to number read
         168
         189,0,128
         145,207
         200
         189,0,129
         145,207
         232
         ~~138~~
         ~~201,12~~
         144,229
         165,205   150
         141,5,212
         165,204
         141,4,212
         76,98,228
```

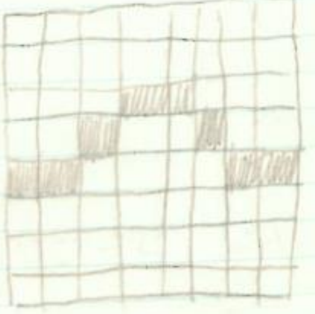
100
116
120
130
140

```
POKE 204,0 POKE 205,0
POKE 207,0
POKE 208,50
F.X=128*256 TO 128*256+11
POKE X,50
F.X=96+107 TO 129*256+11
POKE
```

13
10

If dont have pirate victory or dont have treasure on island then die,





```
IF ES THEN POKE CH+2,0:POKE CH+3,0:POKE CH+4,255:ES=0
IF ES=0 THEN POKE CH+4,195:POKE CH+3,36:POKE CH+2,24:ES=1
```

~~Decrease boundary of islands by one on both top and bottom~~

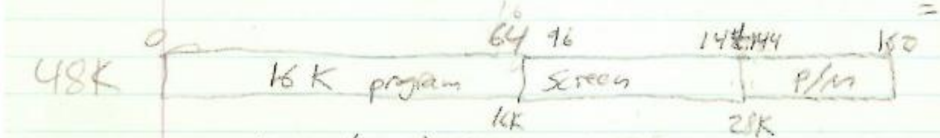
~~Center ship in scrolling screen~~

~~Change border character back to 3 colors~~

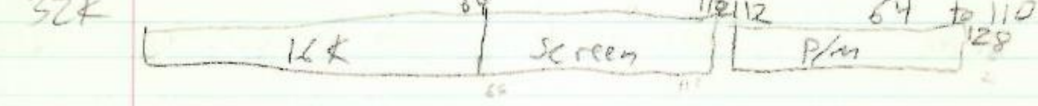
At intermission, after all islands cleared,
show you cashing in money at a bank (western style)

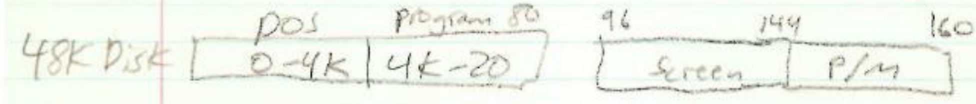
ADD SOUND EffectS
AND PUT BORDER Of
CHARACTER 35 ON SCREEN
ADD POKE MEMORY TO 0 USR
SAVE MEMORY
AT END Of Of GAME POKE MAN off SCREEN

PEEK(106)=160   S=120   clears from S-24 to S+22
= 96 to 142



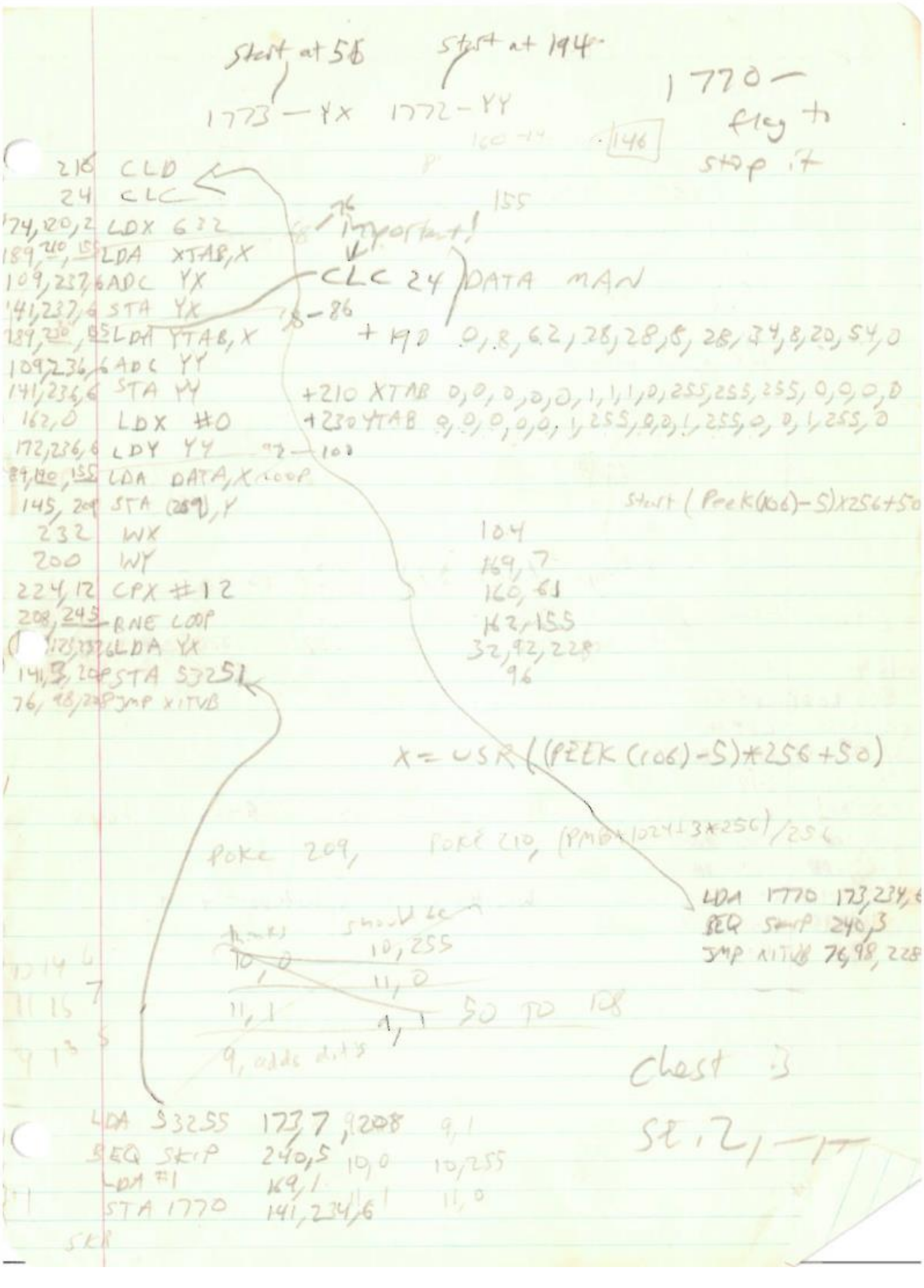
PEEK(106)=128   S=88   clears from 64 to 110





big stick
Circuit up orange white green black brown blue

Atari
Circuit up orange green blue black white brown



8421 9 10 11 7 15

SF — Ship flag

- 0 if you were at an island and <u>can</u> hit ship
- 1 if <u>can't</u> hit ship

ISF — Island flag

- 0 if just at ship and <u>can</u> hit island
- 1 if can't hit island because just at one

if can't hit ship, you have to battle ship, but don't get extra treasure

1740

118
96
22

Change Line 375 232 to 262

Clear memory

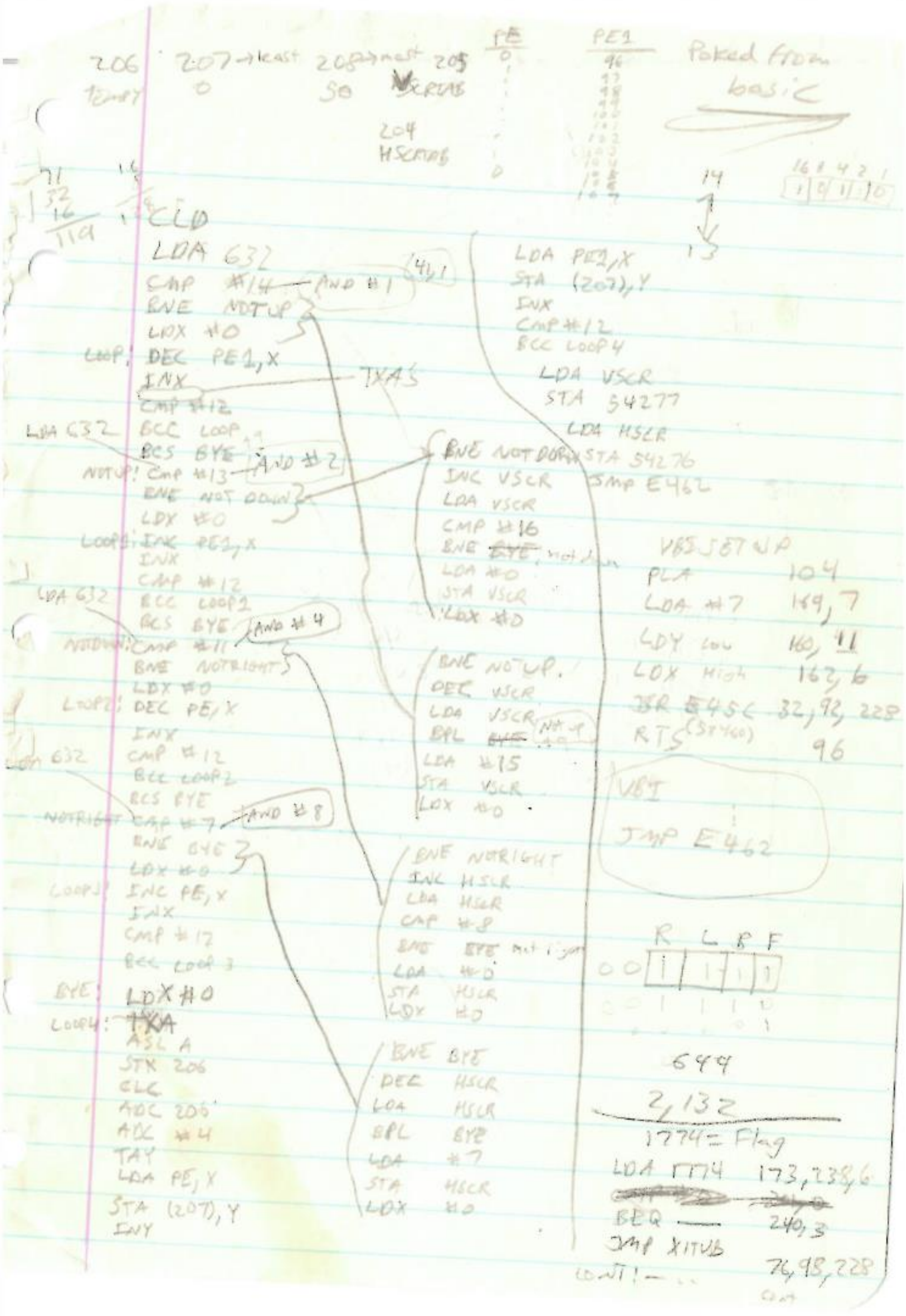
```
104        PLA
169,0 LOOP LDA #0
60,0       LDY #0
145,207 LOOP1 STA (207),Y
200        INY
208,251    BNE LOOP1
230,208    INC 208
165,208    LDA 208
205,233,6  CMP 1769
208,238    BNE LOOP
96         RTS
```

209,210

POKE 207,0 POKE 208,S-24
POKE 1769,S+22

backward branch = 256-number of skips

Whatever to whatever +19



```
206     207→least  208→most  205      PE    PE1
temp    0          58        VSCRTAB  0     96
                                            97
                             204            98
                             HSCRTAB        99
                                            100
                                            101
                                            102
                                            103
                                            104
                                      0     105
                                            106
                                            107
```

Poked from basic

```
71
32
16
119
```

14 ↕ 13

```
16 8 4 2 1
 1 0 1 1 0
```

```
        CLD
        LDA 632
        CMP #14      AND #1   (4,1)
        BNE NOTUP
        LDX #0
LOOP!   DEC PE1,X
        INX                   TXA
        CMP #12
LDA 632 BCC LOOP
        BCS BYE
NOTUP!  CMP #13      AND #2
        BNE NOT DOWN
        LDX #0
LOOP1:  INC PE1,X
        INX
        CMP #12
LDA 632 BCC LOOP1
        BCS BYE      AND #4
NOTDOWN:CMP #11
        BNE NOTRIGHT
        LDX #0
LOOP2!  DEC PE,X
        INX
LDA 632 CMP #12
        BCC LOOP2
        BCS BYE
NOTRIGHT CMP #7      AND #8
        BNE BYE
        LDX #0
LOOP3!  INC PE,X
        INX
        CMP #12
        BCC LOOP 3
BYE!    LDX #0
LOOP4:  ~~TXA~~
        ASL A
        STX 206
        CLC
        ADC 206
        ADC #4
        TAY
        LDA PE,X
        STA (207),Y
        INY
```

```
BNE NOT DOWN
INC VSCR
LDA VSCR
CMP #16
BNE ~~BYE~~ not down
LDA #0
STA VSCR
LDX #0

BNE NOTUP.
DEC VSCR
LDA VSCR
BPL ~~BYE~~   Not up
LDA #15
STA VSCR
LDX #0

BNE NORIGHT
INC HSCR
LDA HSCR
CMP #8
BNE BYE  not right
LDA #0
STA HSCR
LDX #0

BNE BYE
DEC HSCR
LDA HSCR
BPL BYE
LDA #7
STA HSCR
LDX #0
```

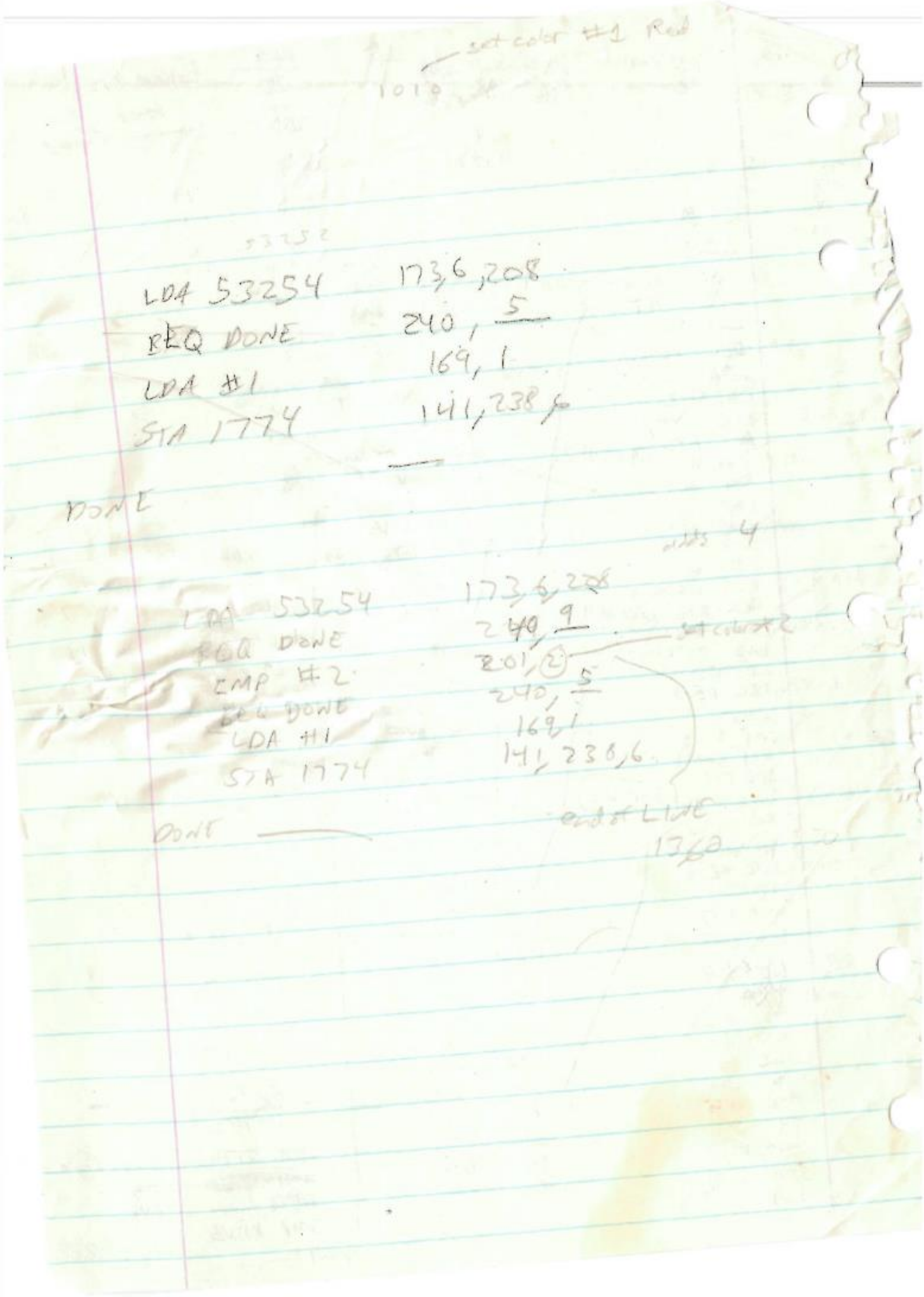
```
LDA PE1,X
STA (207),Y
INX
CMP #12
BCC LOOP4
 LDA VSCR
 STA 54277
  LDA HSCR
  STA 54276
  JMP E462
```

```
VBI SETUP
PLA          104
LDA #7       169,7
LDY Low      160,41
LDX High     162,6
JSR E45C     32,92,228
RTS (58460)  96
```

```
VBI

JMP E462
```

```
   R   L  B  F
00 1   1  1  1
00 1   1  1  0
```

```
  644
2,132
1774= Flag
LDA 1774     173,238,6
BEQ —        240,3
JMP XITVB    76,98,228
CONT!— ..
```



set color #1 Red

1010

```
53252
LDA 53254      173,6,208
BEQ DONE       240, 5
LDA #1         169, 1
STA 1774       141,238,6

DONE
                                adds 4

    LDA 53254      173,6,208
    BEQ DONE       240,9         set color #2
    CMP #2         201,2
    BEQ DONE       240,5
    LDA #1         169,1
    STA 1774       141,238,6

DONE                              end of LINE
                                    1360
```

skip
8
Border
1
251→2

LINE 170 LINE 1290

from page in 208
to page in 1764

Clear Memory except for 2 certain numbers

```
          PLA                104
     LOOP LDY#0              160,0

    LOOP1 LDA (207),Y        177,207
                             201,40 240,17
          CMP #__            201,36
          BEQ CONT           240,8
          CMP #__            201,155
          BEQ CONT           240,4
          LDA #0             169,0
          STA (207),Y        145,207
     CONT INY                200
          BNE LOOP1          208,239 235
  Cmp #40
  BEQ CONT
          INC 208            230,208
          LDA 208            165,208
          CMP 1764           205,233,6
          BNE LOOP           208,228 224
          RTS                96
```

256
17
239

256
28
228

R=1
L=2
G=3
HS=4
T=9

FRE=19222 now
FRE=19451 then

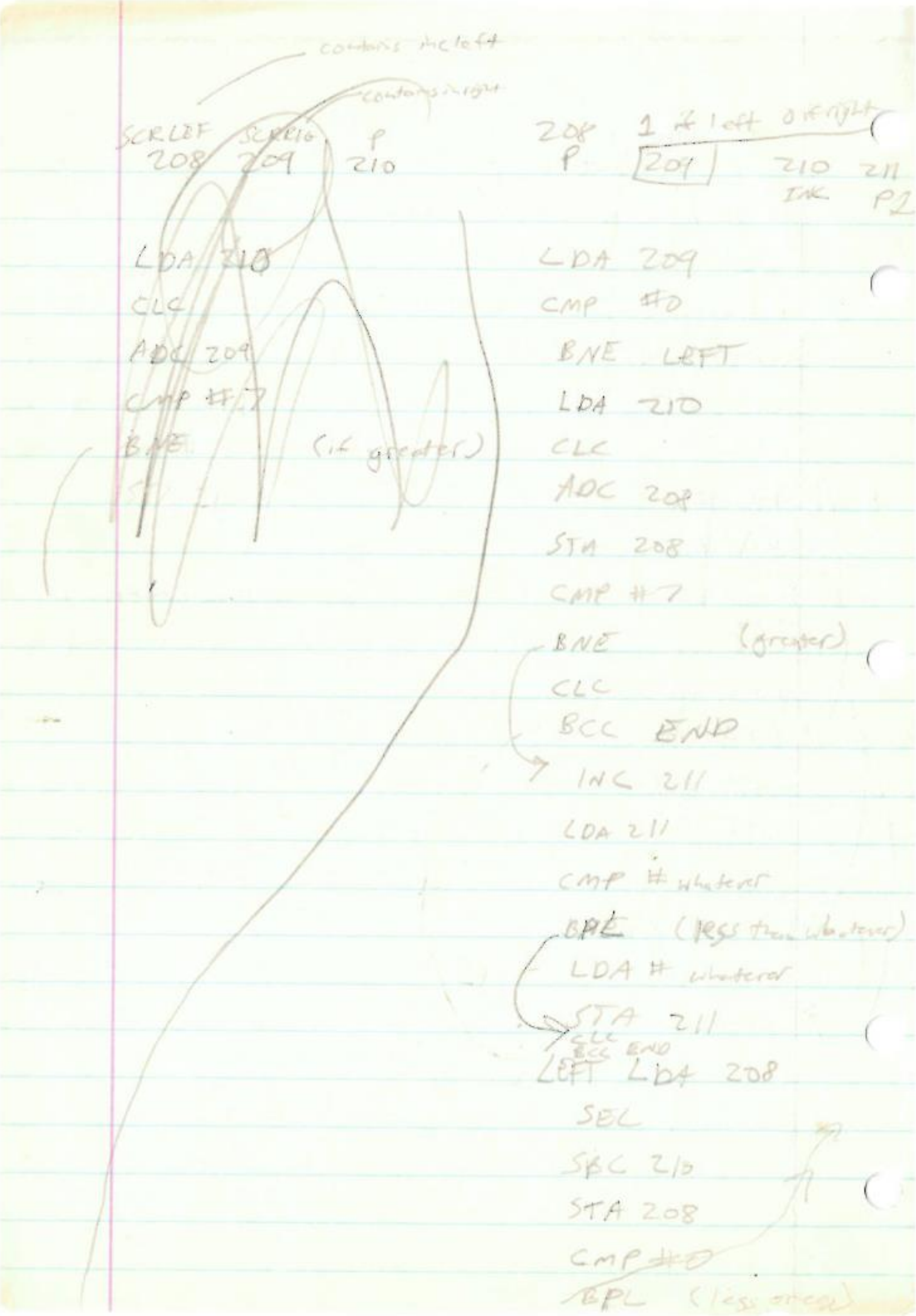
451
222
229 bytes

FRE=19474 ends up

whatever to whatever + 29

```
FOR SS=(S-U)*K TO (S-U)*K+8448 STEP 4:
IF PEEK(SS)=40 THEN POKE SS,40:N.SS

170 IF HIT=A1 OR HIT=R THEN 490
```

```
    Inc  P  P1
    208, 209, 210

LDA 208
CLC
ADC 209
CMP #7
BNE            (branch if greater than 7 )
INC 210
LDA 210
CMP # whatever   (branch if greater than whatever)
BNE
CLC
BCC
LDA #0
STA 201
CLC
BCC
LDA #  whatever
STA 210
LDA
[illegible]
```



contains the left

contains right

```
SCRLEF  SCRRIG   P
 208     209    210

LDA 210
CLC
ADC 209
CMP #7
BNE          (if greater)
```

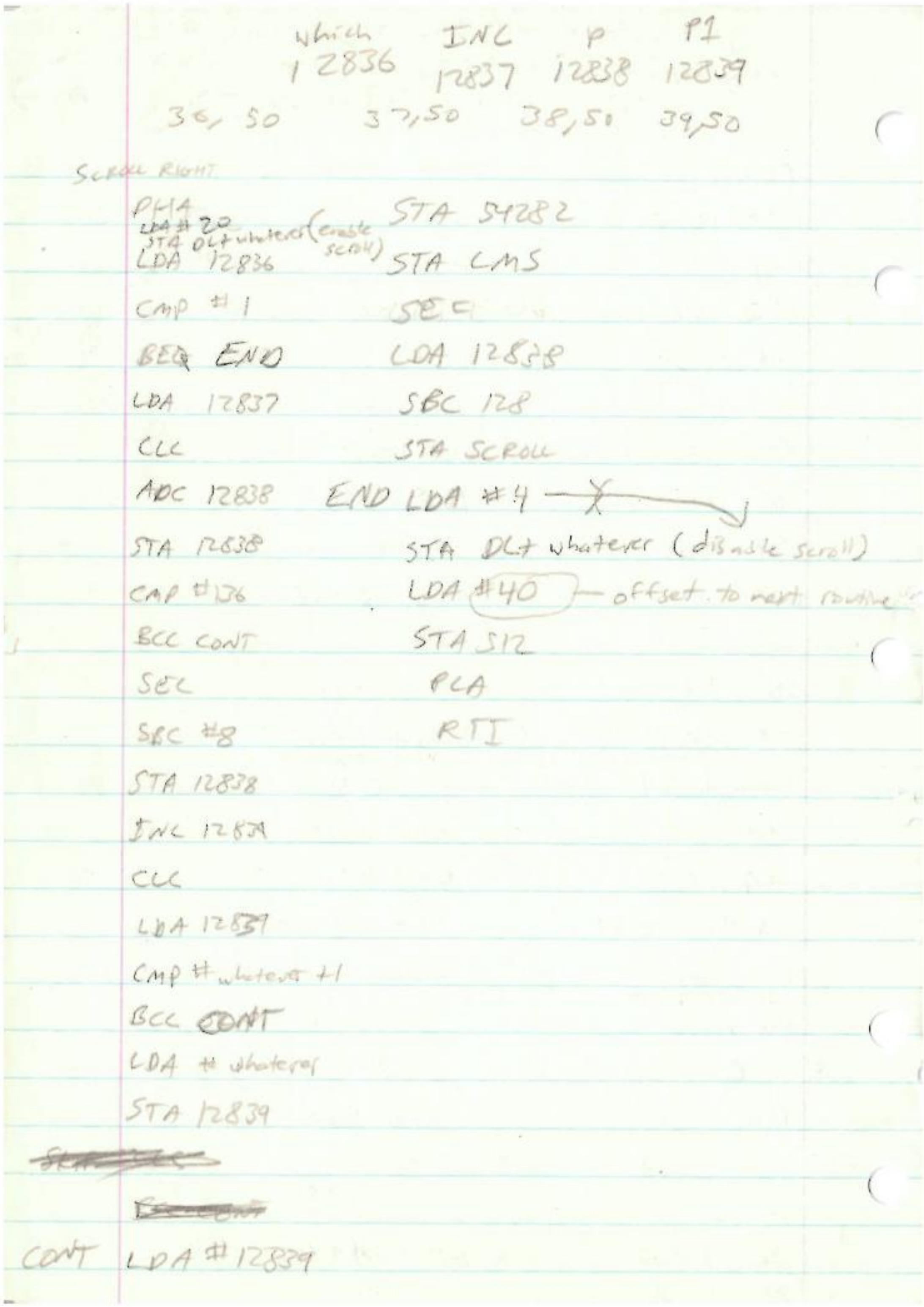
```
208   1 if left 0 if right
P     209       210   211
                INK   P2

      LDA 209
      CMP #0
      BNE  LEFT
      LDA  210
      CLC
      ADC  208
      STA  208
      CMP #7
      BNE          (greater)
      CLC
      BCC  END
      INC 211
      LDA 211
      CMP # whatever
      BPL   (less than whatever)
      LDA # whatever
      STA  211
      CLC
      BCC END
LEFT  LDA  208
      SEC
      SBC 210
      STA 208
      CMP #0
      BPL   (less or equal)
```

If this doesn't work, then I will have to use many separate interrupts (12) one for each six lines to scroll left, 2 to scroll right. In each, enable scroll by poke scroll capability to each line, On back

```
     PHA
     LDA 12836
     CMP #1
     BEQ LEFT
     LDA 12837
     CLC
     ADC 12838
     STA 12838
     CMP #136
BEQ  BCC CONT
     SEC
     SBC #8
     STA 12838
     INC 12839
     CLC
     LDA 12839
     CMP# whatever +1
     BCC SKIP
     LDA # whatever
     STA 12839
SKIP CLC
     BCC CONT
LEFT LDA 12838
     SEC
     SBC 12837
     STA 12838
```

```
        12429
        CLC
        CMP # 121
        BCS CONT
        ADC #8
        STA 12838
        DEC 12839
        LDA 12839
        CMP # whatever
        BCS CONT
        LDA # whatever
        STA 12839
CONT    LDA 12839
        STA 54282
        STA LMS
        LDA 12836
        CMP #1
        BEQ NEWLEFT
        SEC
        LDA 12838
        SBC#128
        STA SCROLL
        CLC
        BCC END
NEWLEFT LDA #128
        SEC
```

```
    SBC 12838
    STA SCROLL
END
    LDA #40   0 For
    STA 512
    PLA
    RTI
```



which	INC	p	P1
12836	12837	12838	12839
36, 50	37,50	38,51	39,50

SCROLL RIGHT

```
      PHA
      LDA #20
      STA DL+whatever (enable scroll)
      LDA 12836
      CMP #1
      BEQ END
      LDA 12837
      CLC
      ADC 12838
      STA 12838
      CMP #136
      BCC CONT
      SEC
      SBC #8
      STA 12838
      INC 12839
      CLC
      LDA 12839
      CMP #whatever +1
      BCC CONT
      LDA # whatever
      STA 12839
CONT  LDA #12839
      STA 54282
      STA CMS
      SEC
      LDA 12838
      SBC 128
      STA SCROLL
END   LDA #4
      STA DL+whatever (disable scroll)
      LDA #40 — offset to next routine
      STA S12
      PLA
      RTI
```

PUT 1 into 204 originally

~~LDA LMSVAL,X~~

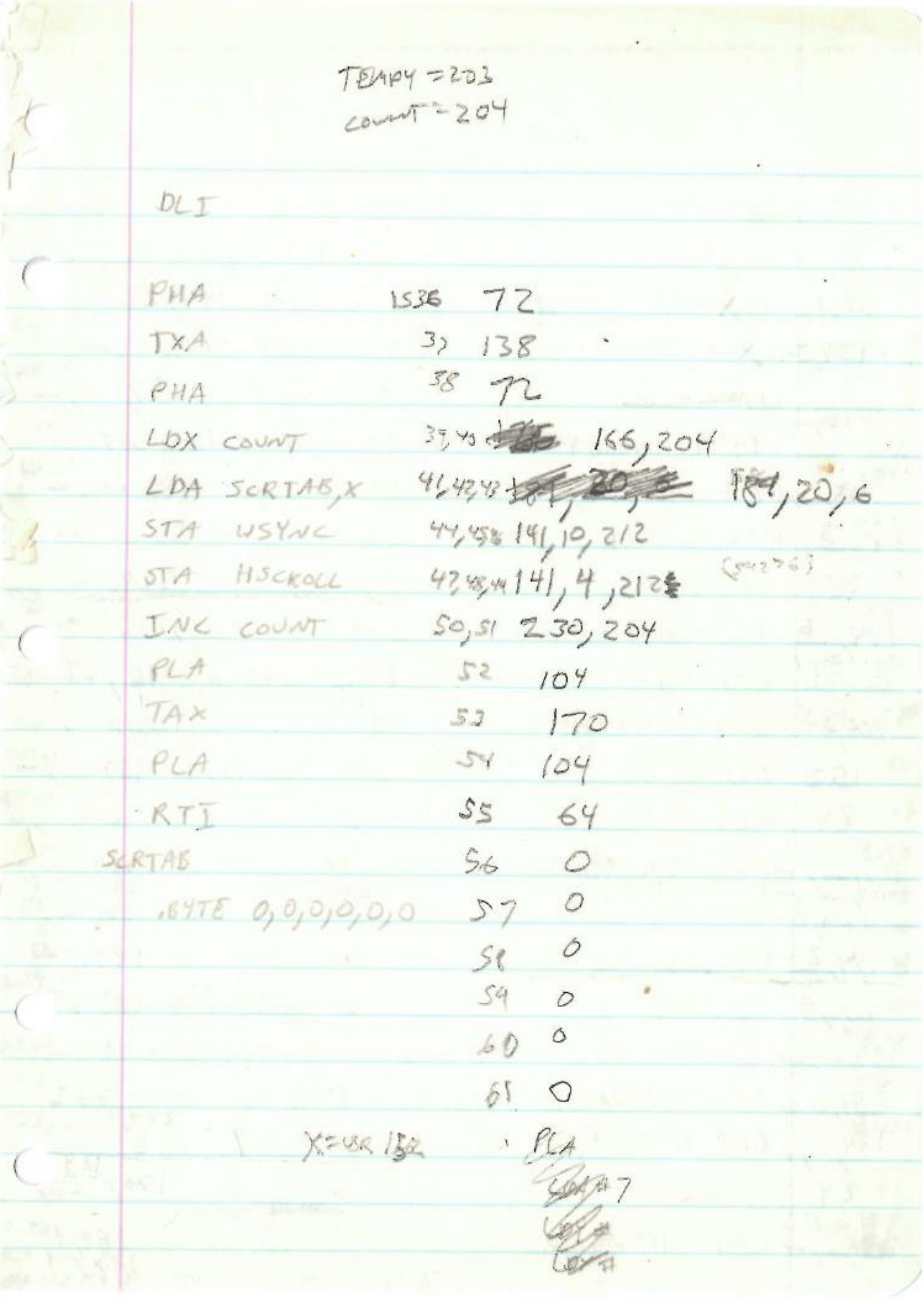
```
LDA  LMSVAL
STA  whatever
LDA  LMSVALHI
STA  whatever
```

SCRTAB=140,6
HOWMUCH=146,6
WHATDO=152,6
LMSVAL=158,6
HIVAL=164,6
LOVAL=170,6

Tables

	current val							
	SCRTAB	HOWMUCH	WHATDO	LMSVAL		HIVAL		LOVAL
676 1	4	82 1 1	88 1 0	94 1		1700 1	254	06 1
77 2	4	83 2 1	89 2 0	95 2		01 2	254	07 2
78 3	4	84 3 1	90 3 0	96 3	122	02 3		08 3
79 4	4	85 4 1	91 4 0	97 4		03 4		09 4
80 5	4	86 5 1	92 5 0	98 5		04 5		10 5
81 6	4	87 6 1	93 6 0	94 6		05 6		1711 6

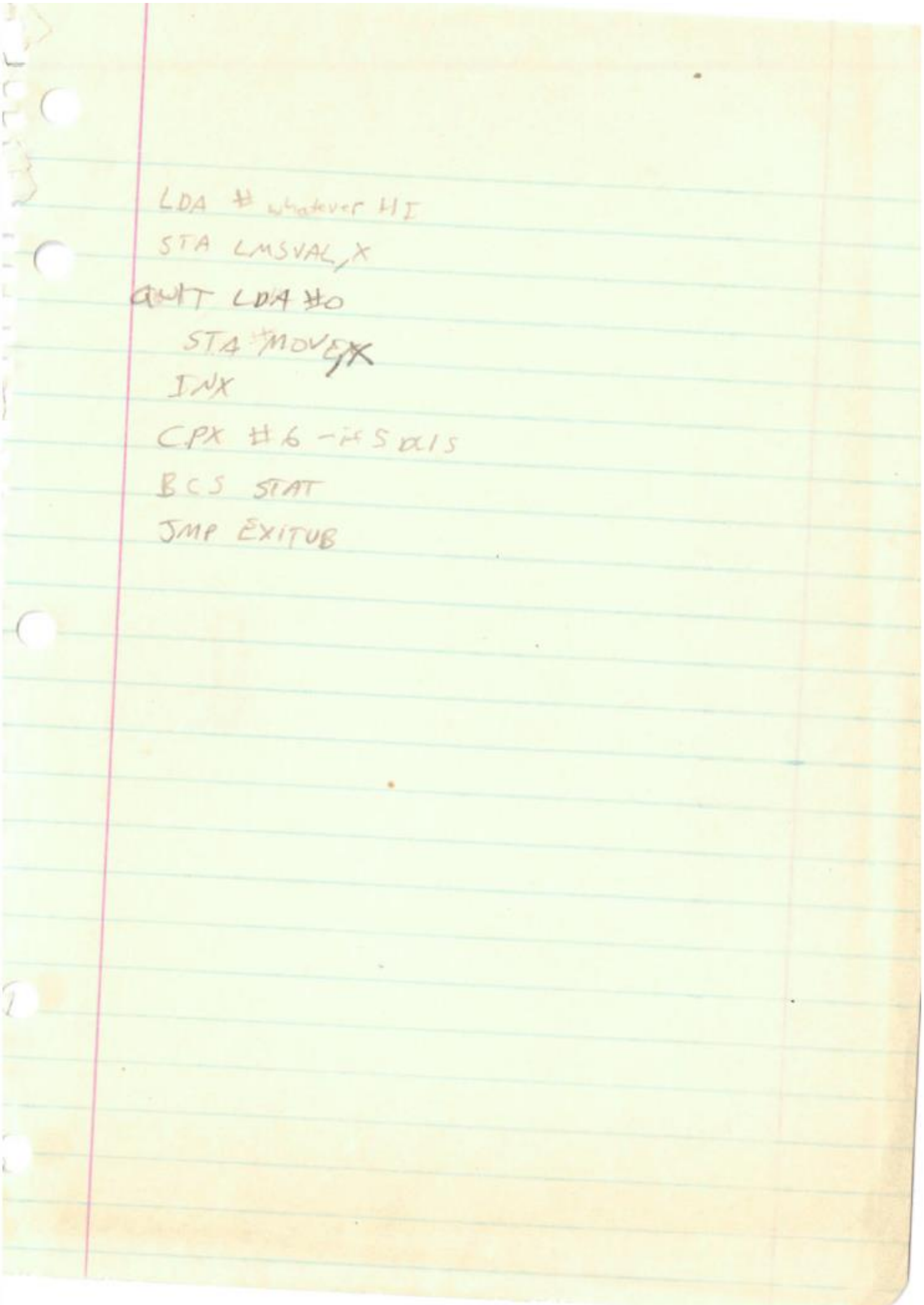
```
62        104         PLA
63,64     169,7       LDA #7
65,66     160,37      LDY #
67,68     162,6       LDX #
69,70,71  32,92,228   JSR SETVB
72        96          RTS
73,74     169,1       LDA #2   DLI
75,76     133,204     STA 204
77,78,79  76,98,228   JMP EXTVB
```



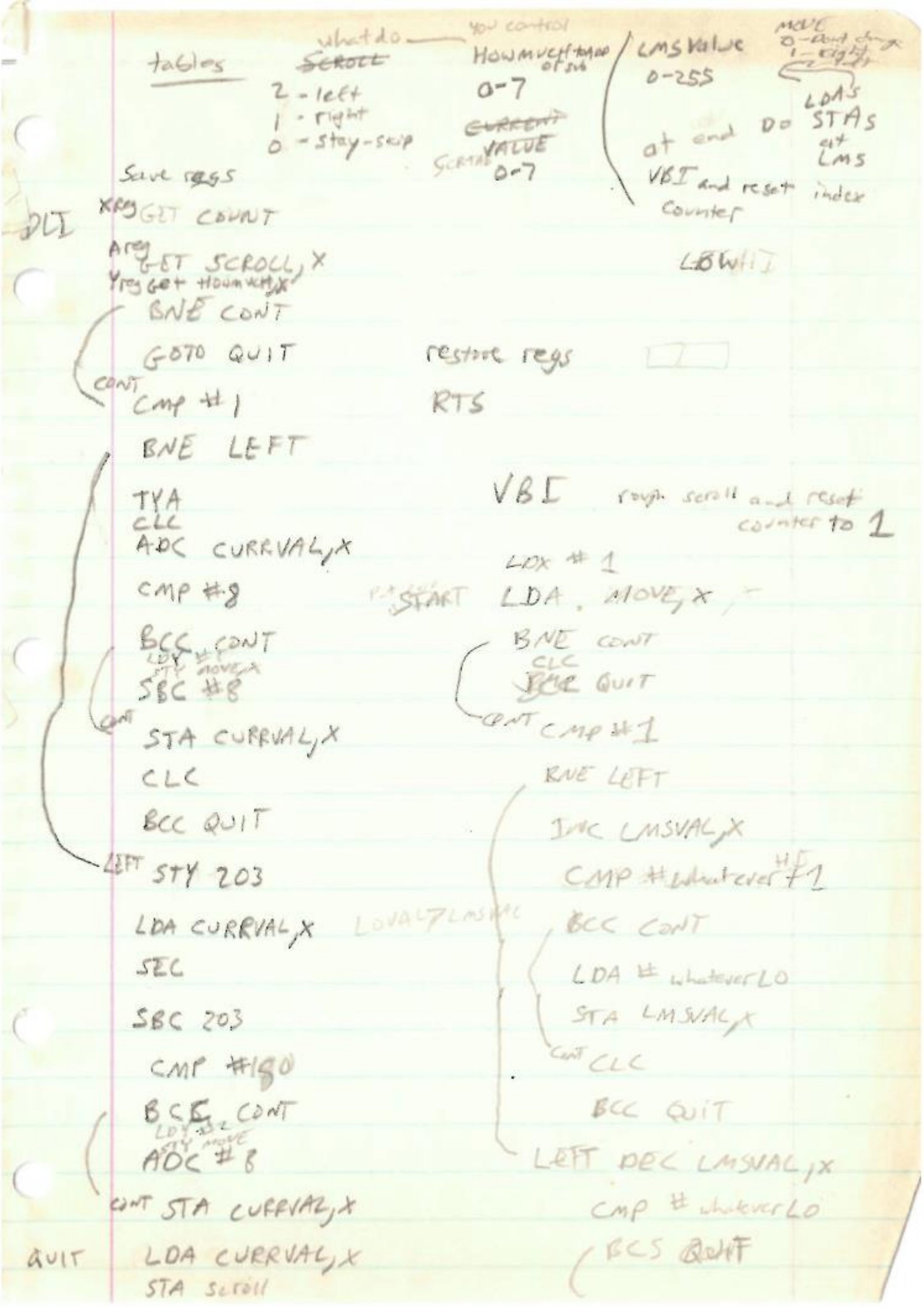
```
              TEMPY =203
              COUNT =204

DLI

PHA                 1536  72
TXA                   37  138
PHA                   38  72
LDX  COUNT          39,40  166,204
LDA  SCRTAB,X       41,42,43       189,20,6
STA  WSYNC          44,45,46 141,10,212
STA  HSCROLL        47,48,49 141,4,212      (54276)
INC  COUNT          50,51 230,204
PLA                   52   104
TAX                   53   170
PLA                   54   104
RTI                   55   64
SCRTAB                56   0
  .BYTE 0,0,0,0,0,0   57   0
                      58   0
                      59   0
                      60   0
                      61   0

            X=USR 152     PLA
```

```
VBI

73,74     162,1        LDX #1                 4
75,76     134,204      STX COUNT
                       ~~LDX COUNT~~
77,78,79  188,-,-
START                  LDY HOWMUCH,X
80,81,82  189,-,-      LDA WHATDO,X
83,84     208,3        BNE CONT1
85        24           CLC
86,87     144,76       BCC QUIT-CONT3
88,89     201,1
CONT1                  CMP #1
90,91     208,35       BNE LEFT-CONT2
92        152          TYA
93        24           CLC
94,95,96  125,-,-      ADC SCRTAB,X
97,98,99  157,-,-      STA SCRTAB,X
100,101   201,8        CMP #8
102,103   144,23       BCC CONT2
105       233,8        SBC #8
6,07,08   157,-,-      STA SCRTAB,X
9,10,11   254,-,-      INC LMSVAL,X
12,13,14  189,-,-      LDA LMSVAL,X
15        24           CLC
          221 16,17,18
          ~~[illegible]~~,-,-   CMP HIVAL,X
          70
```

```
                                   42
                                   50
      BCC CONT2        144,6        19,20
      LDA LOVAL,X      189,-,-      21,22,23
      STA LMSVAL,X     157,-,-      24,25,26
LEFT
CONT2 STY 203          132,203      27,28
      LDA SCRTAB,X     189,-,-      29,30,31
      SEC              56           32
      SBC 203          229,203      33,34
      STA SCRTAB,X     157,-,-      35,36,37
      BPL CONT3        16,24        38,39
      CLC              24
      ADC #8           105,8        40
                                    41,42
      STA SCRTAB,X     157,-,-      43,44,45
      DEC LMSVAL,X     222,-,-      46,47,48
      LDA LOVAL,X      189,-,-      49,50,51
      CLC              24           52
      CMP LMSVAL,X     221,-,-      53,54,55
      BCC CONT3        144,6        56,57
      LDA HIVAL,X      189,-,-      58,59,60
QUIT
      STA LMSVAL,X     157,-,-      61,62,63
CONT3 INX              232          64
      CPX #7 -if 6 DLIS  224,7      65,66
      BCS START        176,164      67,68
      ~~LDX #1~~   DO LMS
      LDA #1           169,1        69,70
      STA COUNT        133,204      71
73,74,75 JMP EXITVB    76,[illegible]
```



```
 LDA # whatever HI
 STA LMSVAL,X
QUIT LDA #0
  STA #MOVE,X
 INX
 CPX #6 -# Sals
 BCS STAT
 JMP EXITUB
```



tables

what do ~~SCROLL~~
2 - left
1 - right
0 - stay-skip

you control
HowMuch to add of sub
0-7

~~CURRENT~~
SCRTAB VALUE
0-7

LMS Value
0-255

MOVE
0-Don't change
1-right
2-left

at end Do LDA's STAs at LMS
VBI and reset index counter

LBWHI

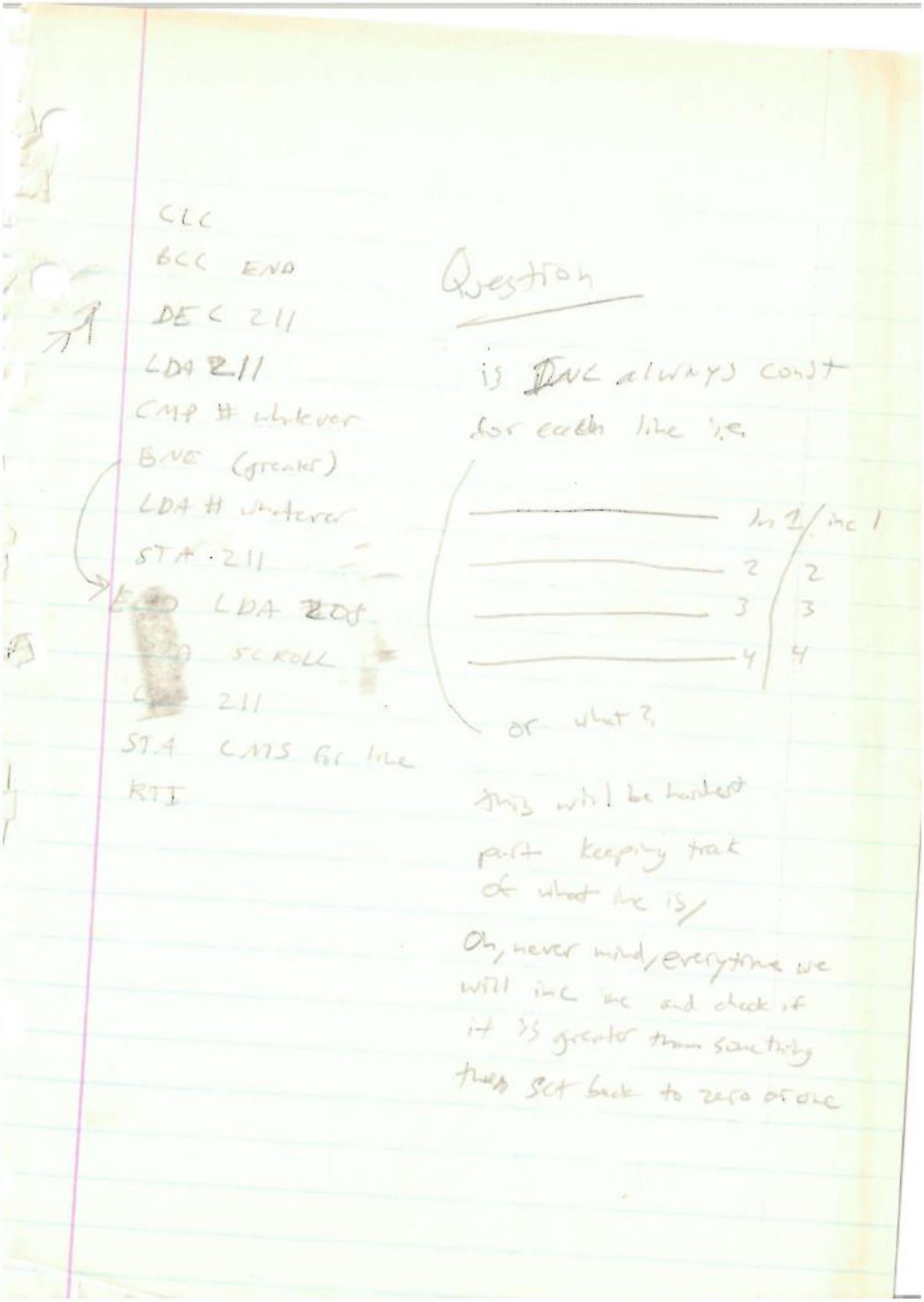
Save regs

```
DLI  XReg GET COUNT
     AReg GET SCROLL,X
     YReg Get HowMuch,X
          BNE CONT
          GOTO QUIT
     CONT CMP #1
          BNE LEFT
          TYA
          CLC
          ADC CURRVAL,X
          CMP #8
          BCC CONT
          LDY #1
          STY MOVE,X
          SBC #8
     CONT STA CURRVAL,X
          CLC
          BCC QUIT
     LEFT STY 203
          LDA CURRVAL,X      LOVAL>LMSVAL
          SEC
          SBC 203
          CMP #180
          BCS CONT
          LDY #2
          STY MOVE
          ADC #8
     CONT STA CURRVAL,X
QUIT      LDA CURRVAL,X
          STA Scroll
```

```
restore regs
RTS
```

VBI rough scroll and reset counter to 1

```
      LDX #1
START LDA MOVE,X
      BNE CONT
      CLC
      ~~BNE~~ BCC QUIT
CONT  CMP #1
      BNE LEFT
      INC LMSVAL,X
      CMP #whateverHI+1
      BCC CONT
      LDA #whateverLO
      STA LMSVAL,X
CONT  CLC
      BCC QUIT
LEFT  DEC LMSVAL,X
      CMP #whateverLO
      BCS QUIT
```



```
CLC
BCC  END
DEC  211
LDA  Z11
CMP  # whatever
BNE  (greater)
LDA  # whatever
STA  .211
END  LDA  20F
STA  SCROLL
LDA  211
STA  CMS for line
RTI
```

Question

is INC always const for each line i.e.

ln 1 / inc 1
2 / 2
3 / 3
4 / 4

or what?

this will be hardest part keeping track of what line is/

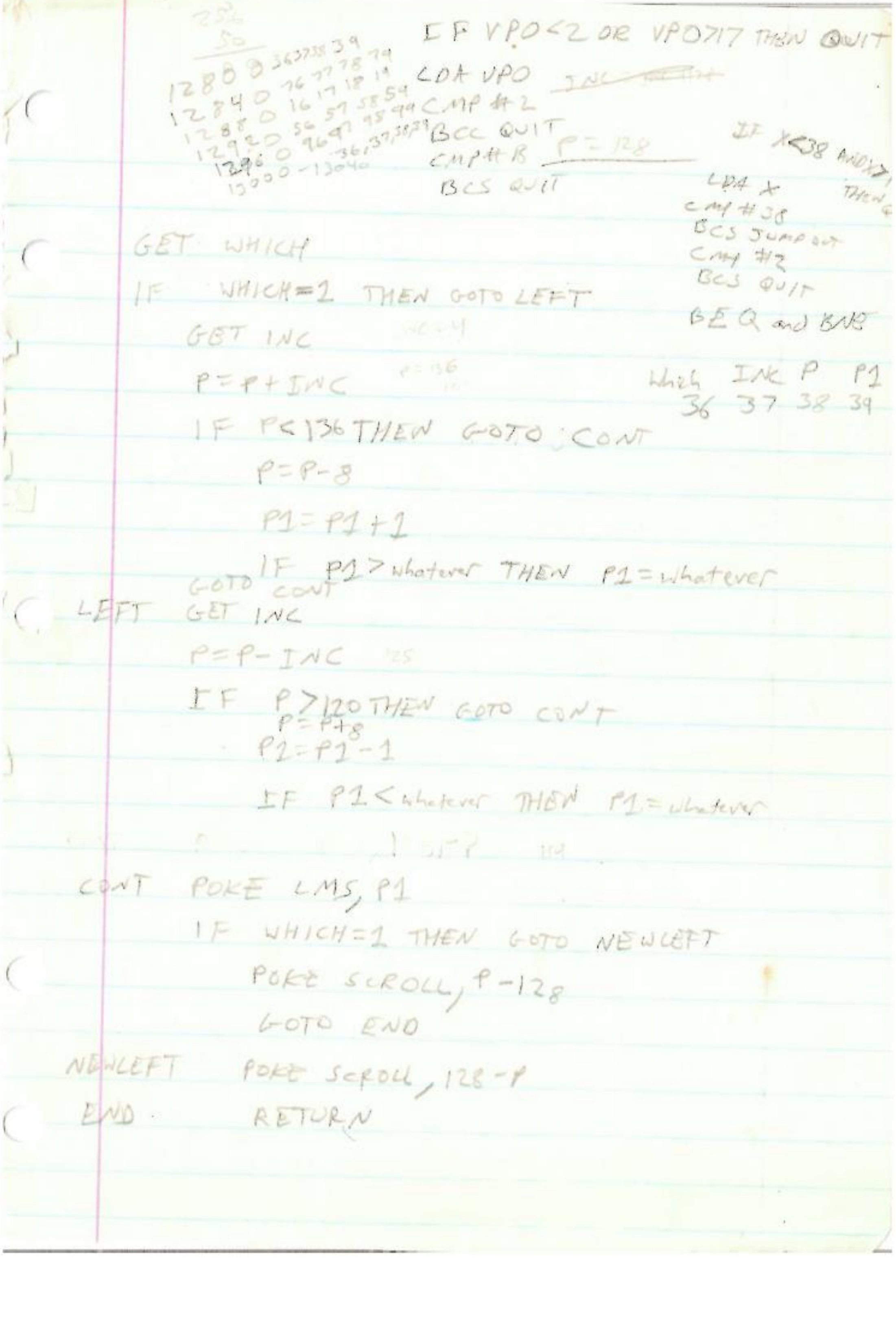
Oh, never mind, everytime we will inc line and check if it is greater than something then set back to zero or one

```
36,50=130,50
37,50=131,50
38,50=132,50
39,50=133,50

15726    110,    61    54276
```

```
72
173,36,50
201,1
240,39
173,37,50
24
109 38,50
141,38,50
201,136
208 BNE 144,60
56
233,8
141,38,50
238,39,50
24
173,39,50
201,72
208 BNE 144,5
169,1 iter
141,39,50
SKIP 24
144,35
LEFT 173,38,50
```

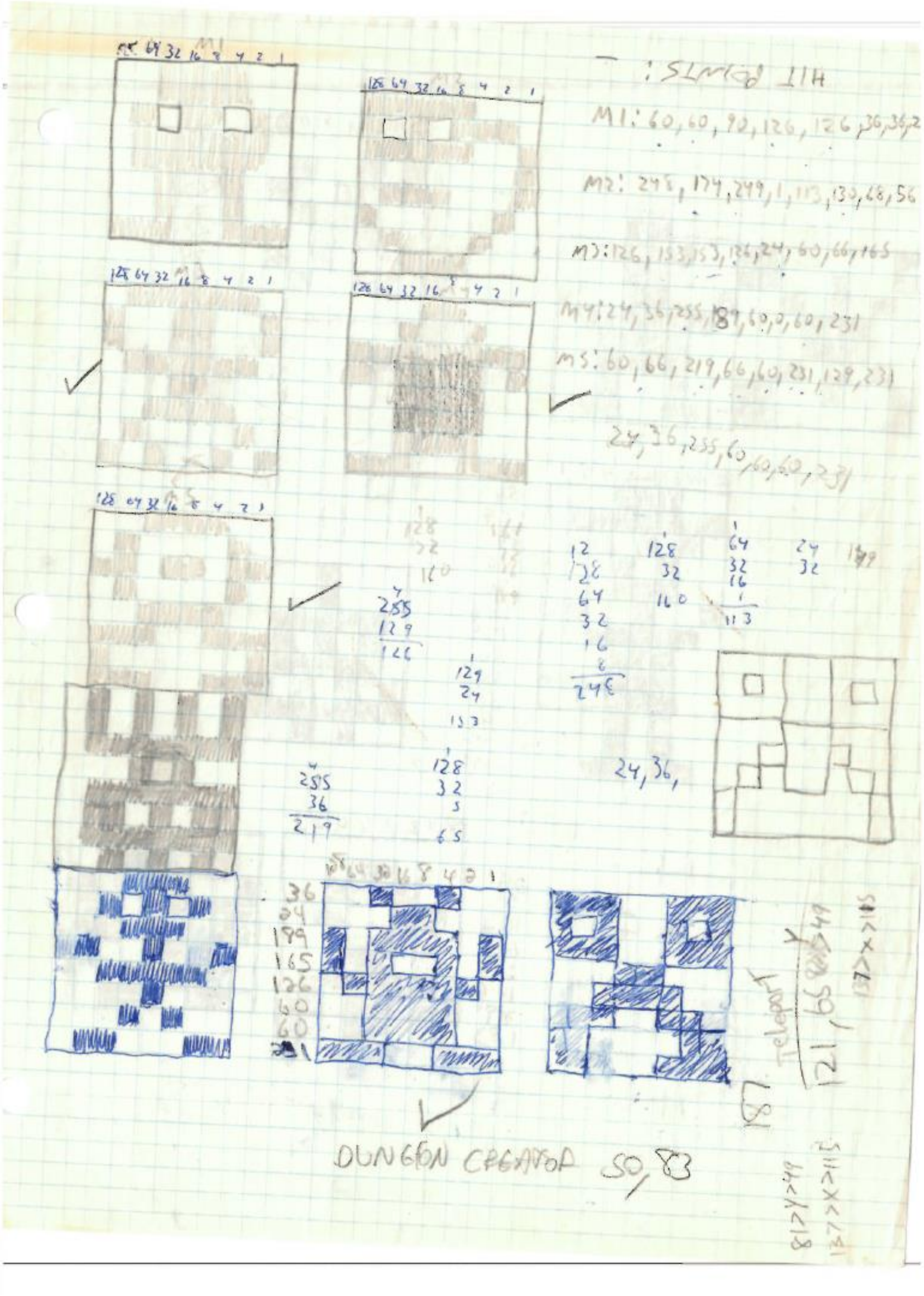
```
56
237,37,50
141,38,50
24
201,121
BEQ 240 176,20
105,8
141,38,50
206,39,50
173,39,50
201,1 iter
BEQ 240 176,5
169,73
141,39,50
CONT 173,39,50
141,10,212
P 141,11,61
141,114,61
173,36,50
201,1
240,12
56
173,38,50
233,128
/50
```

```
141,4,212
24
208 144,9
NEWDOT 169,128
56
237,38,50
141,4,212
END 169,0
141,0,2
104
64
50
79
129
```

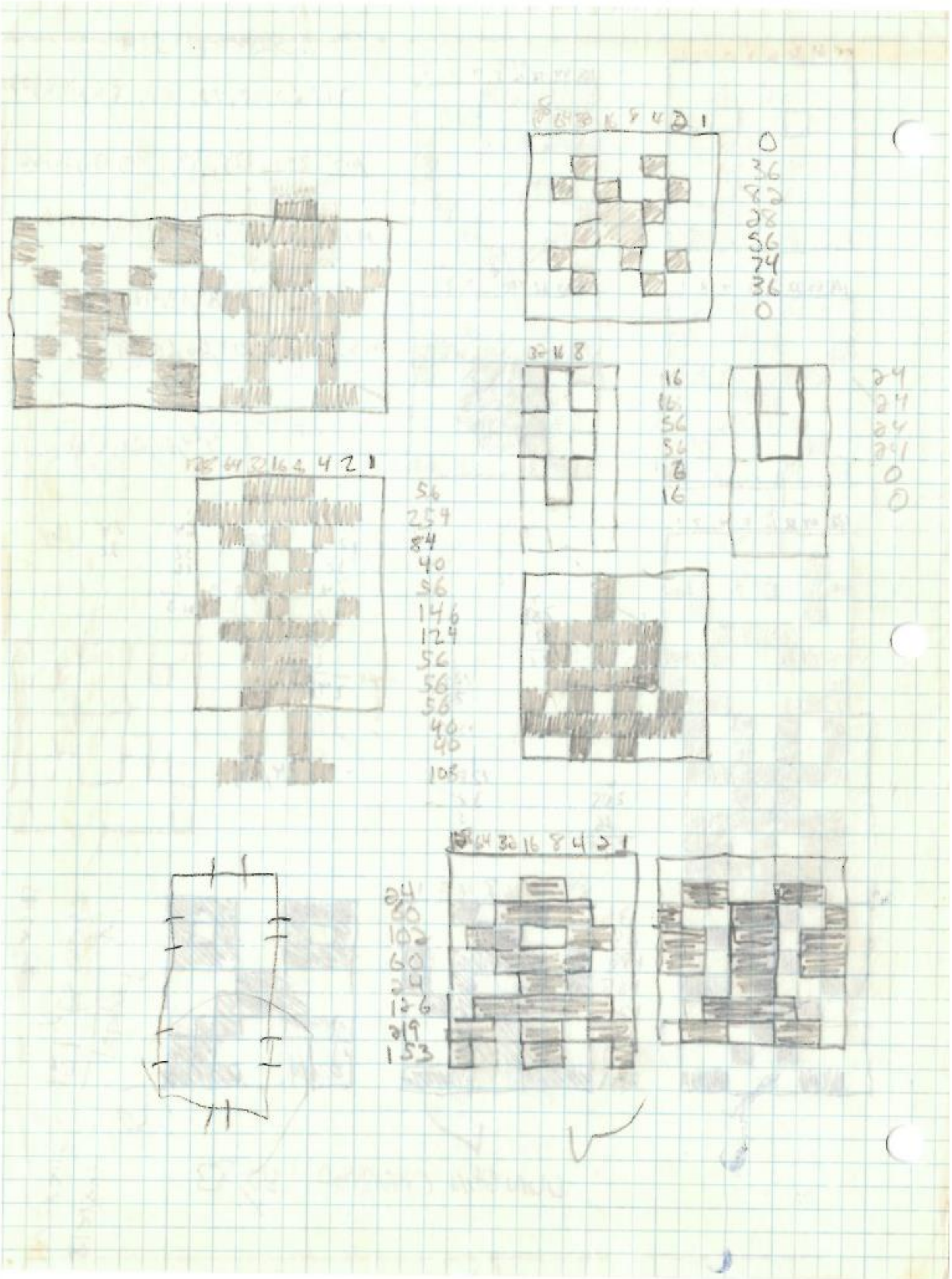


```
256
 50
12800  36 37 38 39          IF VPO<2 OR VPO>17 THEN QUIT
12840  76 77 78 79          LDA VPO      ~~INC~~
12880  16 17 18 19          CMP #2
12920  56 57 58 59          BCC QUIT
12960  96 97 98 99          CMP #18   P=128              IF X<38 AND X>1
13000-13040  36,37,38,39    BCS QUIT                       THEN
                                                    LDA X
                                                    CMP #38
                                                    BCS JUMP OUT
GET WHICH                                           CMP #2
IF  WHICH=1  THEN GOTO LEFT                         BCS QUIT
    GET INC                                         BEQ and BNE
    P=P+INC      P=136                          Which INC P  P1
    IF P<136 THEN GOTO CONT                      36   37 38 39
        P=P-8
        P1=P1+1
        IF P1>whatever THEN P1=whatever
    GOTO CONT
LEFT  GET INC
    P=P-INC   125
    IF P>120 THEN GOTO CONT
        P=P+8
        P1=P1-1
        IF P1<whatever THEN P1=whatever

CONT  POKE LMS, P1
      IF WHICH=1 THEN GOTO NEWLEFT
         POKE SCROLL, P-128
         GOTO END
NEWLEFT   POKE SCROLL, 128-P
END       RETURN
```

ROBOT DUNGEON



128 64 32 16 8 4 2 1
M1
HIT POINTS:
M1: 60, 60, 90, 126, 126, 36, 36, 2
M2: 248, 174, 249, 1, 113, 130, 68, 56
M3: 126, 153, 153, 126, 24, 60, 66, 165
M4: 24, 36, 255, 189, 60, 0, 60, 231
M5: 60, 66, 219, 66, 60, 231, 129, 231
24, 36, 255, 60, 60, 60, 231
36
24
189
165
126
60
60
231
24, 36,
Teleport
121, 65
187
DUNGEON CREATOR 50, 83
81>Y>49
137>X>115


128 64 32 16 8 4 2 1
0
36
82
28
56
74
36
0
32 16 8
16
16
56
56
16
16
24
24
24
24
0
0
128 64 32 16 8 4 2 1
56
254
84
40
56
146
124
56
56
56
40
40
108
128 64 32 16 8 4 2 1
24
60
102
60
24
126
219
153

Room, NAME, Status, SC, R, C, H, XA, XB

3,3,15,15,63,63,255,255

48
15
63

255

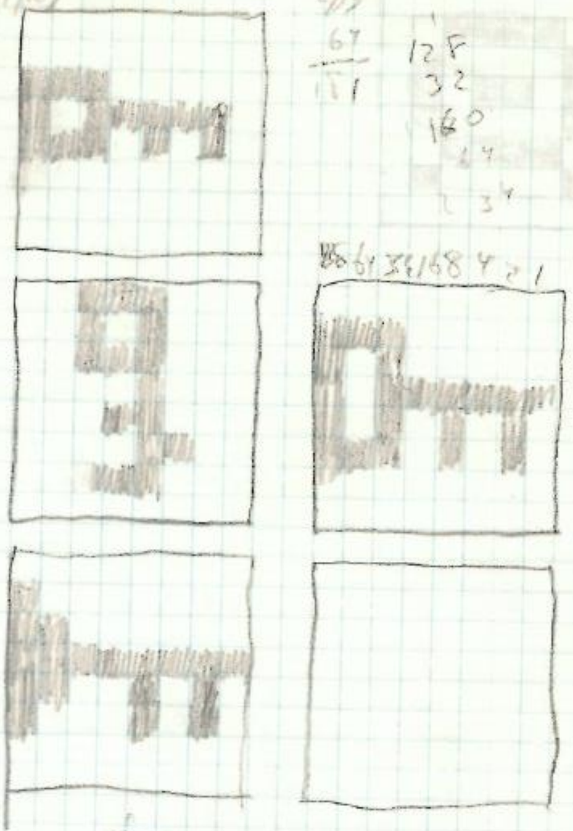
128 64 32 16 8 4 2 1
1 2 4 8 16 32 64 128

192, 192, 240, 240, 252, 252, 255, 255

192
48
30

32
16

Room 187
Score 2490
Hit points -22



255
67
181

12F
32
160
64
34

150
64

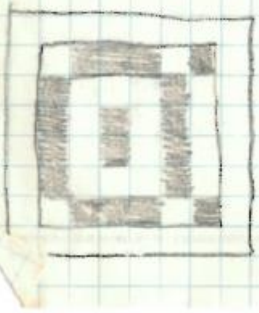
128
96
124

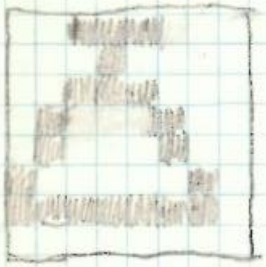
0,0,0,224,60,181,170,234,

224

255
64
181

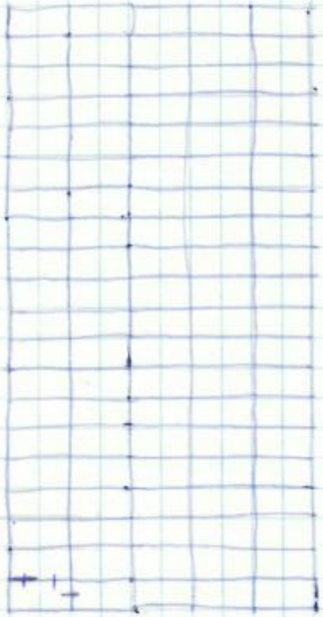
255
64
181

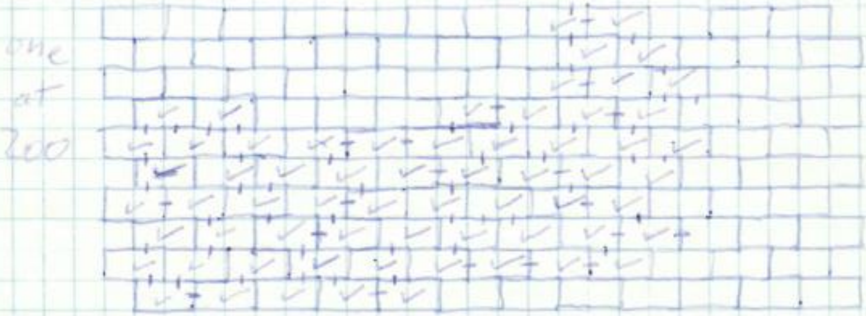




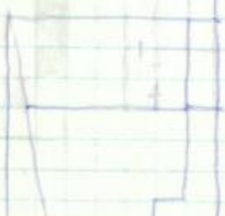
Peek

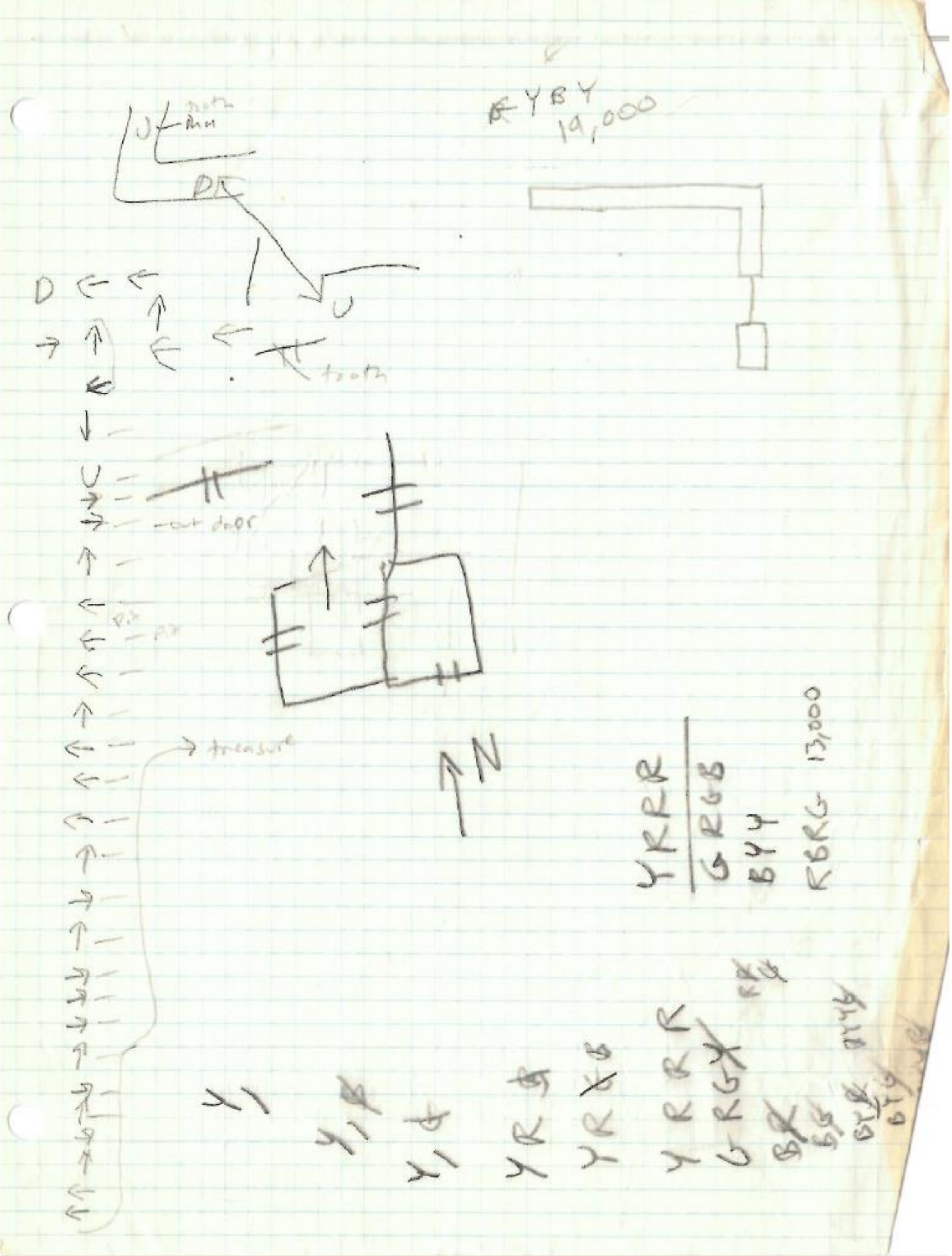
708	=	~~139~~ 136	0,8,11	-wall
709	=	12	1,0,12	-diamond
710	=	148	2,9,4	-monsters
711	=	0	3,9,0	- pit
712	=	70	4,4,6	-background

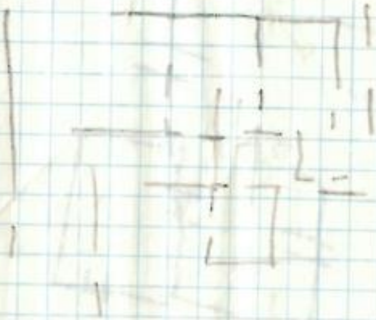


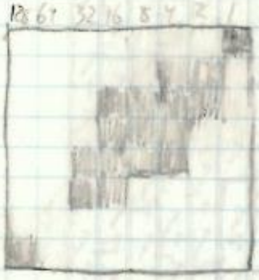


up -128
down -8

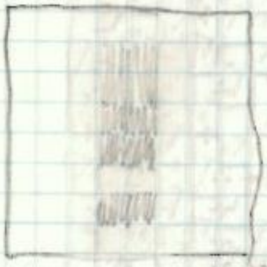



~~R~~YBY
19,000
U
tooth
rm
DE
U
D
tooth
U
out door
p.x
p.x
treasure
N
YRRR
GRGB
BYY
RBRG 13,000
Y1
Y1 ~~B~~
Y1 ~~G~~
YR ~~G~~
YR ~~YB~~
YRRR
GRG~~Y~~
RR
~~Y~~
B~~R~~
BG
B~~R~~G
BYG
BYYG



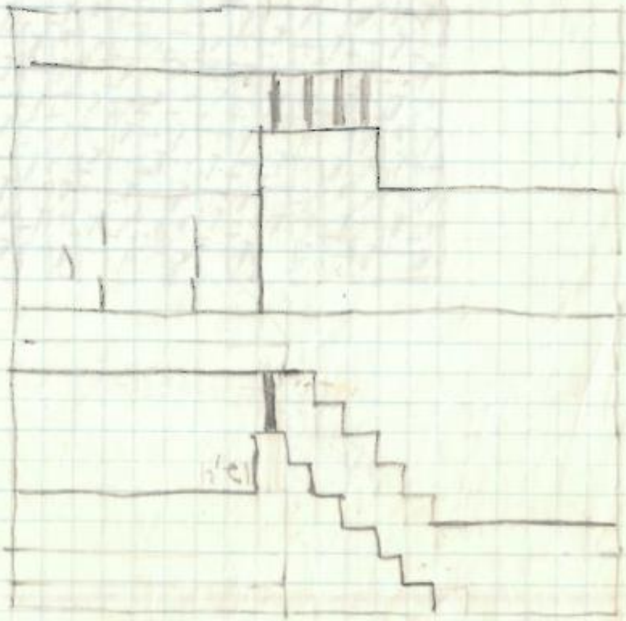


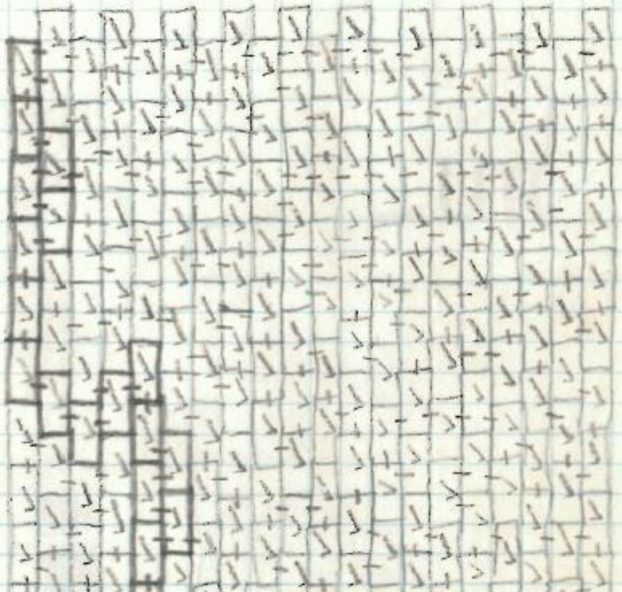
7,7,15,28,56,48,0,128



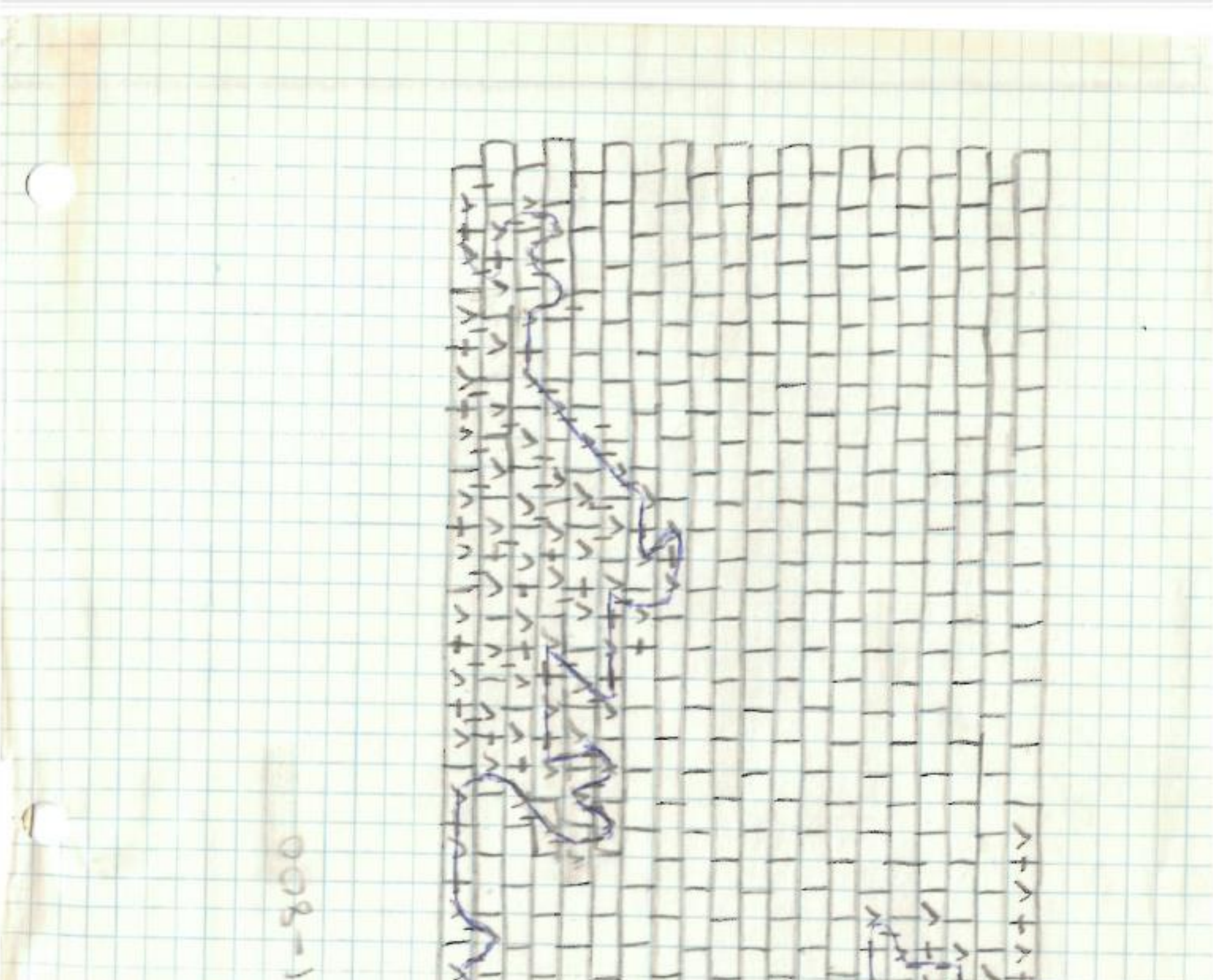
0,24,24,24,24,0,24

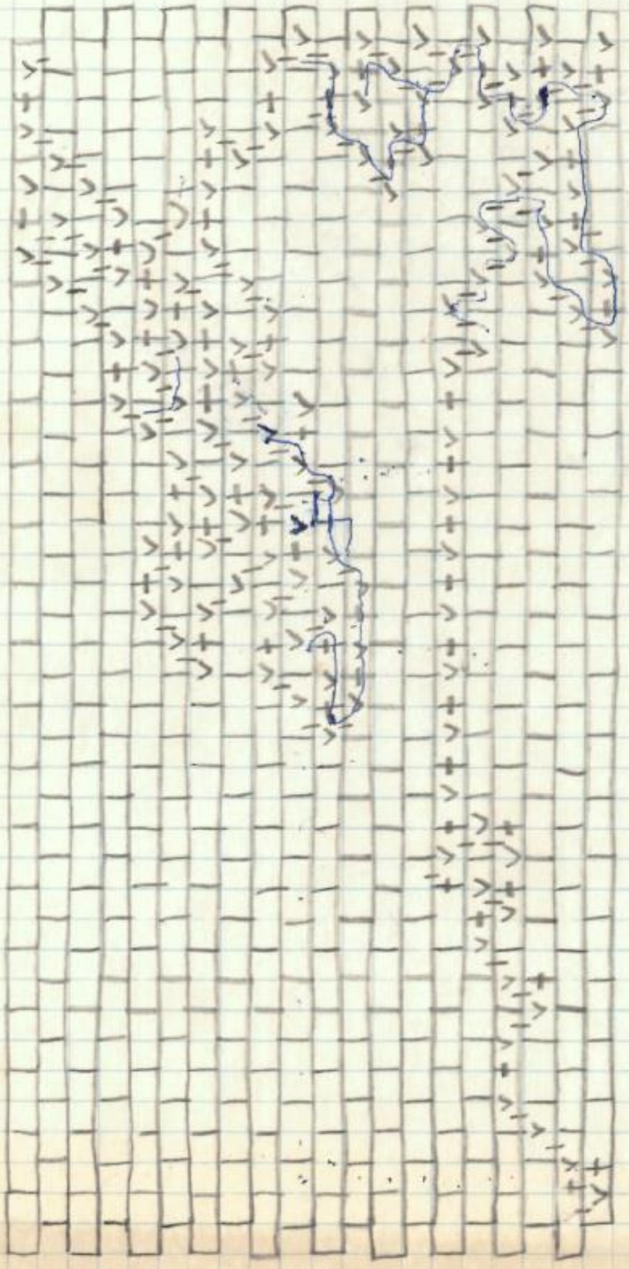
56





1-400

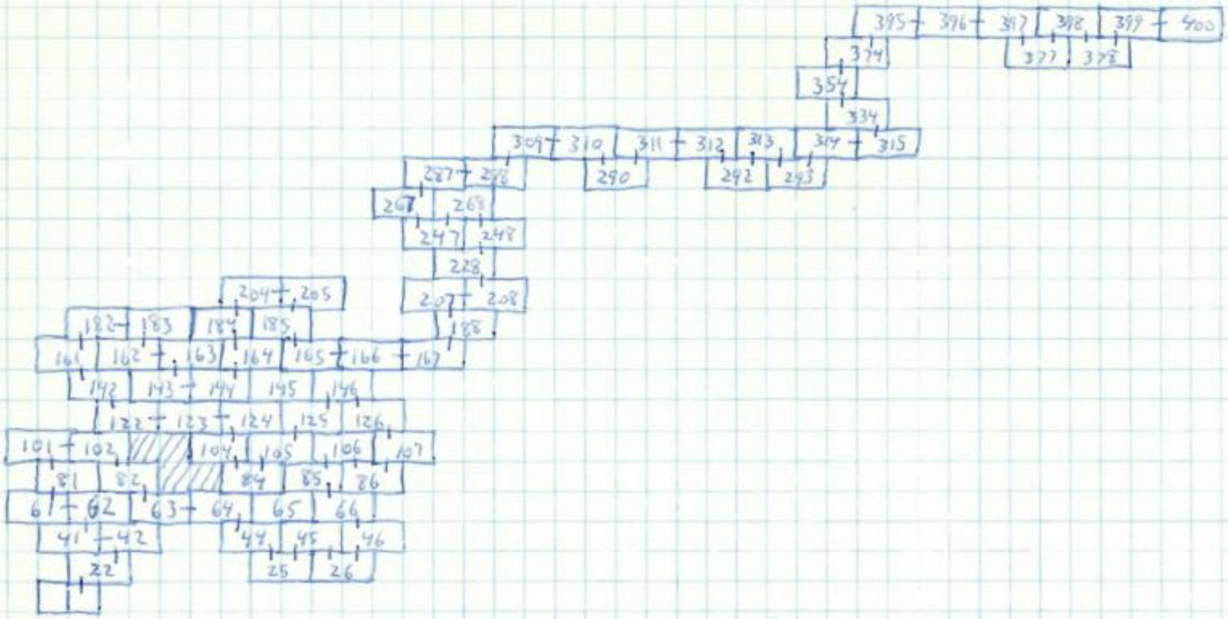

401-800



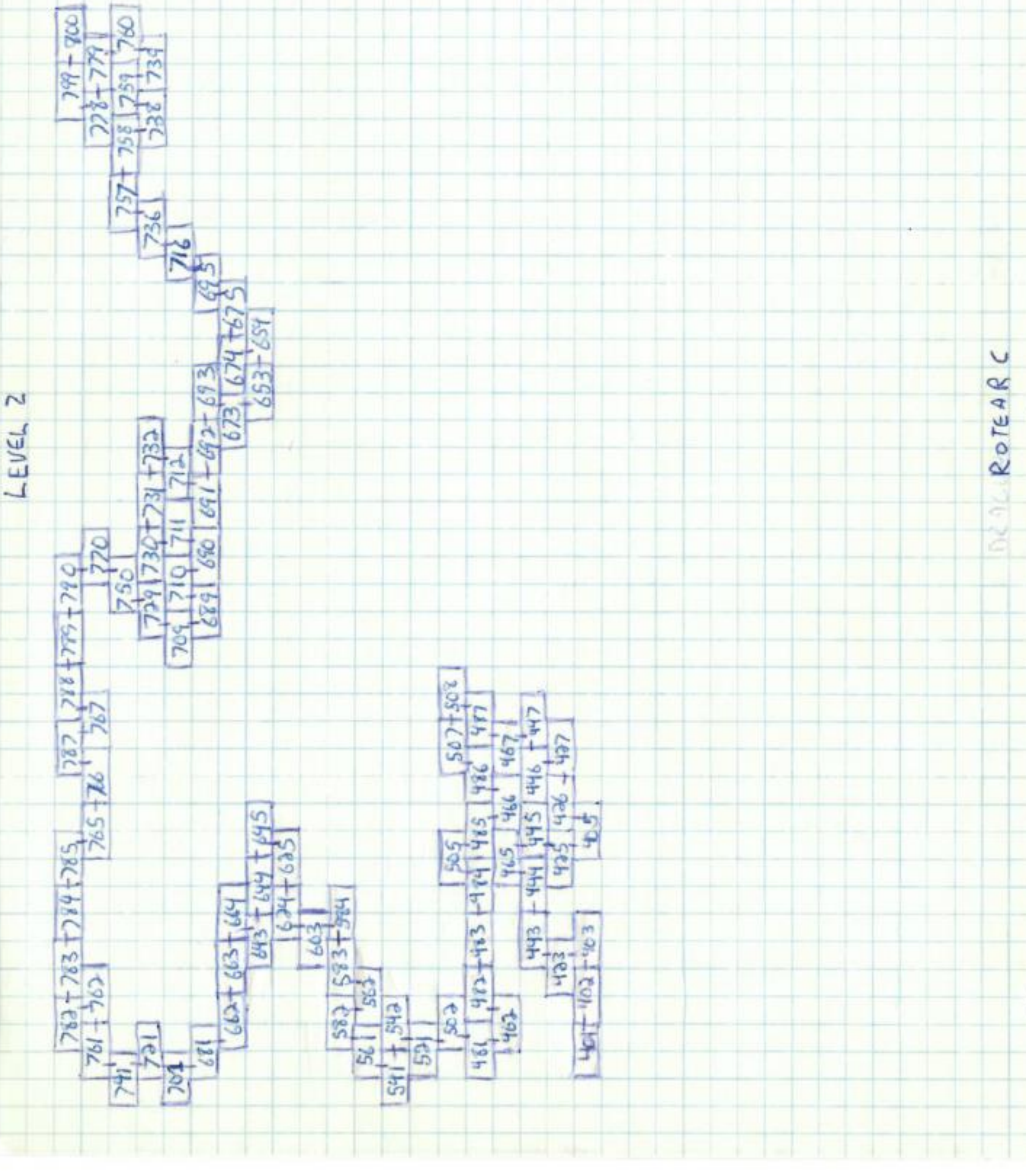
981 – 1200

1012

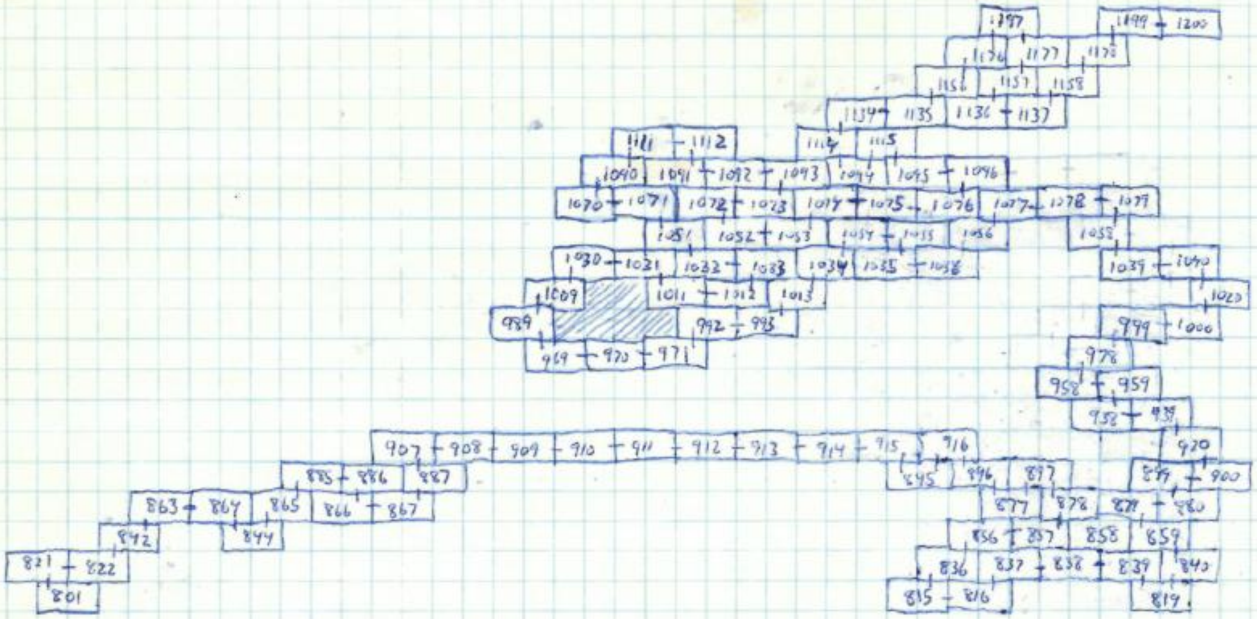
LEVEL I



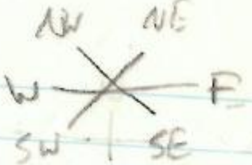
ROTEARC


LEVEL 2
ROTEAR C

LEVEL 3



ROTEARC

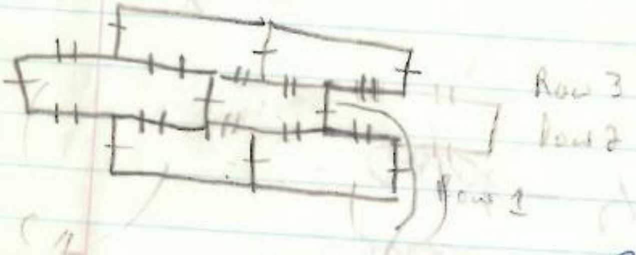


↔ 28 10

↕ 20

8

U	NE	NW	E	D	SE	SW	W
128	64	32	16	8	4	2	1

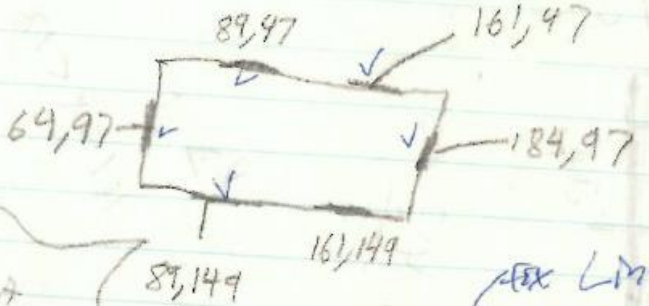


if touching wall and shooting then shot stops

but you will be blown up anyhow

220 W
E
NW
NE
SW
SE

10	14	6
11	15	7
9	13	5



ix
dd
ore
278,0

if you hit a wall then you can't shoot

Sometimes missle X and Y get set to zero

character data is messed up

fix Line 140 add 53278,0 before G.24

79, 80, 81, 82,

83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95

Players

Room Staff

CH

GR

4K wide-screen

Y < 115

161 < X < 176

Life -64

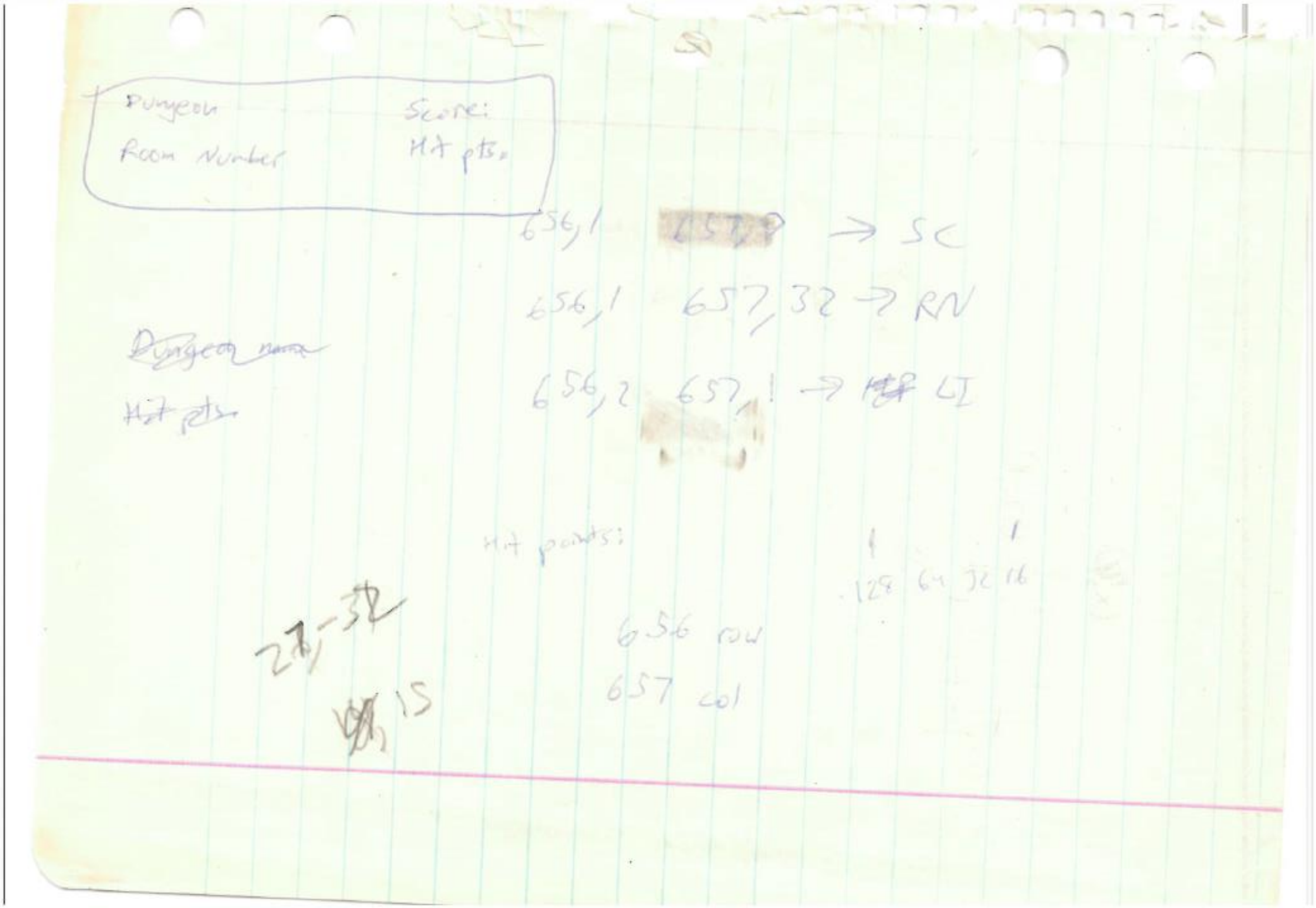
Diamond -32

Silver -16

Gold - 4

X > 177

Y < 66



Dungeon Score:
Room Number Hit pts.

656,1 657,8 → SC

656,1 657,32 → RN

656,2 657,1 → ~~HP~~ LI

Dungeon name

Hit pts.

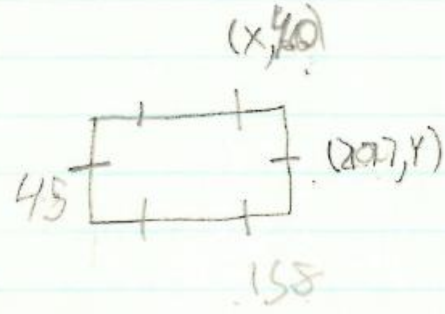
Hit points: 128 64 32 16

656 row

657 col

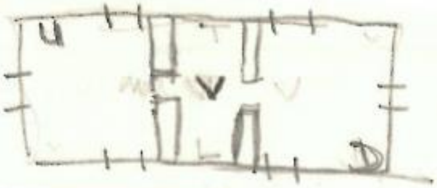
27-32

~~16~~, 15

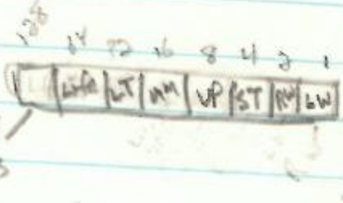


~~R=90~~

R=82 4 86
L=76 N 78
S=83



General Location



magic ([illegible] mirror,
vortex, Life,
~~Lose a Life~~,
treasures

	# per level
magic mirror	25
vortex pool	80
Gain a Life	20
100, 250, 500 treas.	100
1000, 2500, 5000 treas.	25
Left wall	200
Right wall	200.

After ten Lives, a free one acts just like magic mirror.

19308 .
19317

VP { 3, 4, 9, 11, 11, 9, 4, 3
252, 2, 249, 13, 13, 249, 2, 252

J { 112, 248, 112, 32, 0, 0, 0, 0

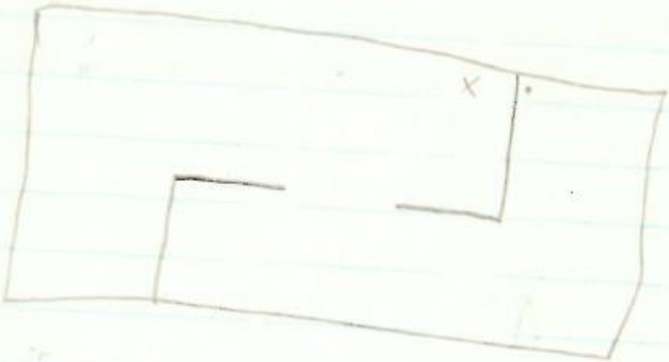
221, 221, 170, 119, 119, 119,
170, 221, 0, 0, 0, 254, 254, 254,
0, 0, 56, 56, 56, 8, 56, 56, 56,
0 ,16, 168, 18, 72, 16, 0, 0, 0
0, 0, 0, 0, 14, 31, 14, 4, 0, 0, 0, 170,
85, 0, 0, 0, 0, 108, 146, 146, 108,
0, 0

3, 4, 9, 11, 11, 9, 4, 3, 252, 2,
249, 13, 13, 249, 2, 252

0
32
128
224
256 6 16 2)492 SC-SC+99 0-20 48 208 0 128 256 128 AA(ZZ)/20 PEEK(53)/208

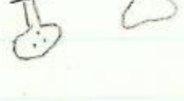
```
   SC=PEEK(88)+256*PEEK(89):M(0)=10:M(1)=11:M(2)=12:M(3)=13:M(4)=
10 FOR ZZ=0 to 4: IF PEEK(203)<(AA(ZZ)*8)+48
   THEN AA(ZZ)=AA(ZZ)-1*(PEEK(SC+BB(ZZ)*20+(AA(ZZ)-1))=0)
20 AA(ZZ)=AA(ZZ)+1 *(PEEK(SC+BB(ZZ)*20+(AA(ZZ)+1))=0 : IF PEEK(1781) <
   (BB(ZZ)*16)+32 THEN BB(ZZ)=BB(ZZ)-1*(PEEK(SC+(BB(ZZ)-1)*20)
30 BB(ZZ)=BB(ZZ)+1 *(PEEK(SC+(BB(ZZ)+1)*20)=0 : NEXT ZZ
40 FOR ZZ=0 TO 4 : POKE SC+BB(ZZ)*20+AA(ZZ), M(ZZ)
```

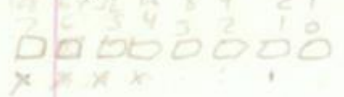
0-9



```
10 DIM AA(4),BB(4),M(4),OA(4),OB(4):M(0)=42: M(1)=43
   M(2)=44: M(3)=45: M(4)=46
20 FOR ZZ=0 TO 4 : OA(ZZ)=AA(ZZ): OB(ZZ)=BB(ZZ):
   SS=INT(RND(0)*2)+1:ON SS GOTO 30,50
30 IF PEEK(203)<(AA(ZZ)*8)+48 THEN AA(ZZ)=AA(ZZ)-1:
   GOTO 70
40 AA(ZZ)=AA(ZZ)+1:GOTO 70
50 IF PEEK(1700)<(BB(ZZ)*16)+32 THEN BB(ZZ) =
   BB(ZZ)-1 : GOTO 70
60 BB(ZZ)=OB(ZZ+1)
           COLOR 3
70 LOCATE AA(ZZ),BB(ZZ),CC : IF CC<>0 THEN
   POSITION OA(ZZ),OB(ZZ) : ?#6;CHR$(M(ZZ))
80 POSITION AA(ZZ),BB(ZZ):?#6;CHR$(M(ZZ))
```

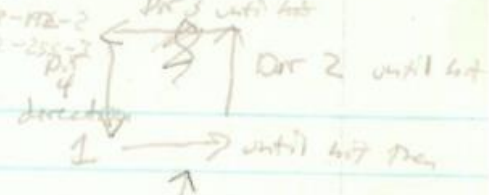
206





70
×12
40
700
240

0-64-0
64-128-1
128-192-2
192-256-3



6 ASR'S ~~Habic~~

~~15~~

You are player shot is player

monsters are characters approx 5

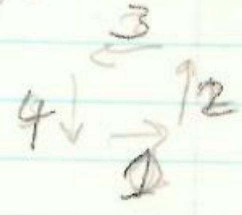
10	11	12	13	14	15
*	+	/	_	.	/

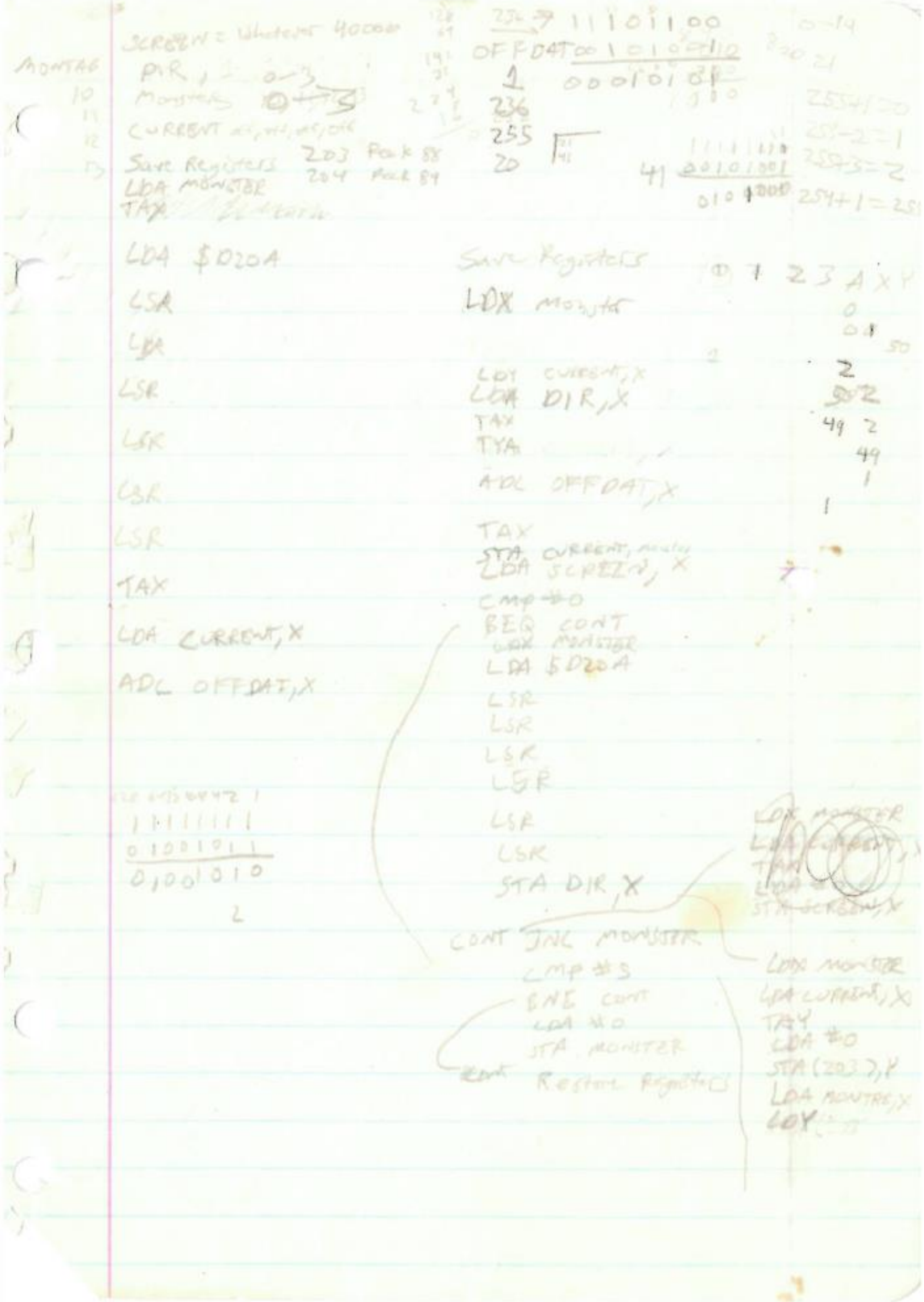
Initialize dir monster 1 to 1 mon 2 to 2 etc

Get random 0-255 divide by 64 then 0-4 add 1

```
monster  get monster number
DL$      maybe use offset to get internal code
         Get direction to move
         ~~change random 1-4 maybe though [illegible]~~
         read offset to add to current location
         add to current
         check if location =0 then move to new location
            if <70 increment dir. to make move in other direction  >4 → 1
         inc monster number if >4 → 1
```

```
you      get number from mem
move     if number=0 then set =1 → bye
DL$        set number= 0
         Get joystick
           Use joystick to getXoffset, y offset
```





```
MONTAG
10
11
12
13

SCREEN = Whatever 4000
DIR , 0-3
Monsters 0 → 3
CURRENT
Save Registers     203  Pock 88
LDA MONSTER        204  Pock 89
TAX
```

```
128
64
191
31
2 7 4
1

236→    1 1 1 0 1 1 0 0
OFFDAT  0 0 1 0 1 0 1 1 0
1       0 0 0 1 0 1 0 1

236
255
20     21
       41

        1 1 1 1 1 1 1 0
41      0 0 1 0 1 0 0 1
        0 1 0 1 0 0 0 0
```

```
0-14
21

255+1=0
255-2=1
255-3=2
254+1=255
```

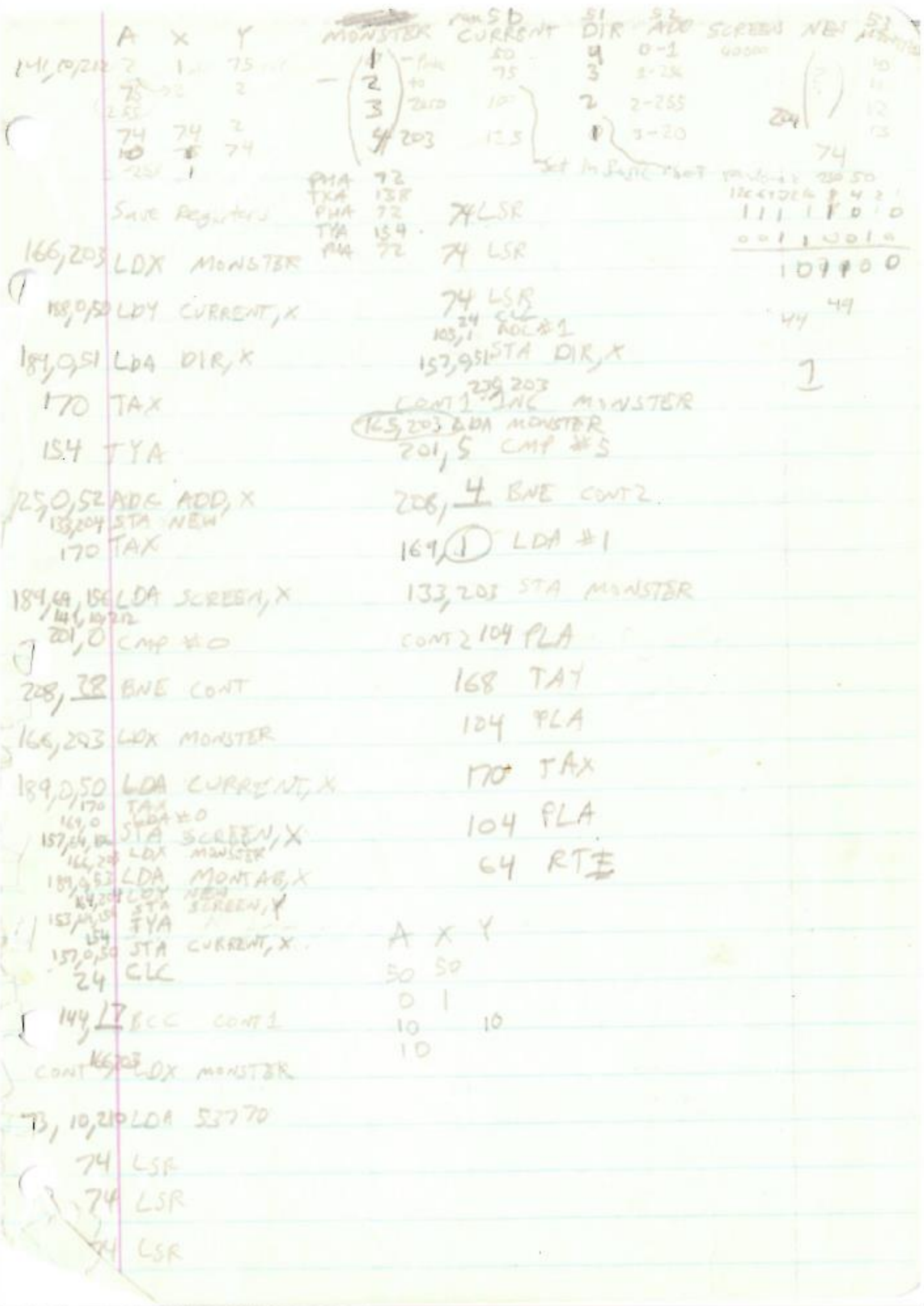
```
LDA $D20A
LSR
LSR
LSR
LSR
LSR
LSR
TAX
LDA CURRENT,X
ADC OFFDAT,X
```

```
128 64 32 16 8 4 2 1
 1  1  1  1 1 1 1 1
 0  1  0  0 1 0 1 1
 0  1  0  0 1 0 1 0

2
```

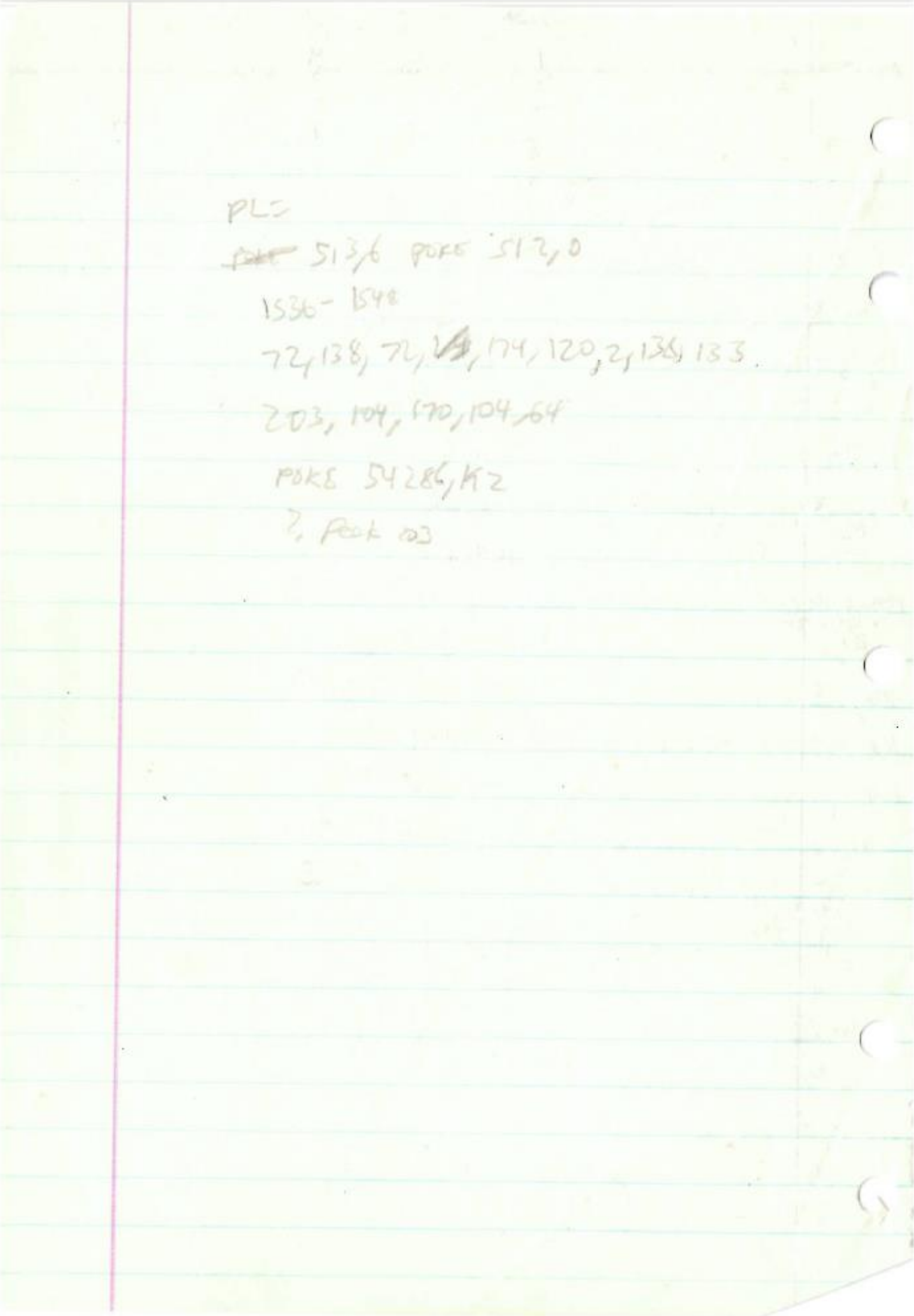
```
     Save Registers
     LDX Monster

     LDY CURRENT,X
     LDA DIR,X
     TAX
     TYA
     ADC OFFDAT,X

     TAX
     STA CURRENT,Monster
     LDA SCREEN,X
     CMP #0
     BEQ CONT
     LDX MONSTER
     LDA $D20A
     LSR
     LSR
     LSR
     LSR
     LSR
     LSR
     STA DIR,X

CONT INC MONSTER
     CMP #3
     BNE CONT
     LDA #0
     STA MONSTER
END  Restore Registers
```

```
0 1 2 3 A X Y
          0
          0 1
              50
          2
          50 2
          49 2
             49
             1
          1
```

```
LDX MONSTER
LDA CURRENT,X
TAX
LDA #0
STA SCREEN,X

LDX MONSTER
LDA CURRENT,X
TAY
LDA #0
STA (203),Y
LDA MONTRE,X
LDY
```



```
           A   X   Y       MONSTER        50        51    52
                                          CURRENT   DIR   ADD     SCREEN   NEW  53 MONTAB
141,10,212 2   1   75      1 -Pha         50        4     0-1     40000         50
           75      2     - 2  to          75        3     1-256                 51
           255             3  Zero        100       2     2-255                 52
           74  74  2       4  203         125       1     3-20     204          53
           10  1   74                                                            74
           255 1                 Set in Basic that                  256 50
                           PHA  72                              128 64 32 16 8 4 2 1
                           TXA  138                              1  1  1  1  1 0 1 0
     Save Registers        PHA  72      74 LSR                   0  0  1  1  0 0 1 0
                           TYA  154                                 1  0  1  1 0 0
                           PHA  72      74 LSR
166,203  LDX  MONSTER                                               44   49
                                        74 LSR
168,0,50 LDY  CURRENT,X                 24 CLC
                                   105,1   ADC #1                    1
185,0,51 LDA  DIR,X              157,0,51  STA  DIR,X
    170  TAX                          238,203
    154  TYA                     CONT1  INC  MONSTER
                                 165,203 LDA MONSTER
125,0,52 ADC  ADD,X                201,5  CMP #5
133,204  STA  NEW                  208,4  BNE CONT2
    170  TAX                       169,1  LDA #1
185,64,156 LDA SCREEN,X          133,203  STA  MONSTER
141,10,212                     CONT2 104  PLA
  201,0  CMP #0                      168  TAY
 208,38  BNE  CONT                   104  PLA
166,203  LDX  MONSTER                170  TAX
185,0,50 LDA  CURRENT,X              104  PLA
    170  TAX                          64  RTI
  169,0  LDA #0
157,64,156 STA SCREEN,X
166,203  LDX  MONSTER
185,0,53 LDA  MONTAB,X
164,204  LDY  NEW                  A   X   Y
153,64,156 STA SCREEN,Y            50  50
    154  TYA                       0   1
157,0,50 STA  CURRENT,X            10      10
     24  CLC                       10
 144,17  BCC  CONT1
CONT 166,203 LDX MONSTER
173,10,210 LDA 53770
     74  LSR
     74  LSR
     74  LSR
```



PL=

~~Poke~~ 513,6 POKE 512,0

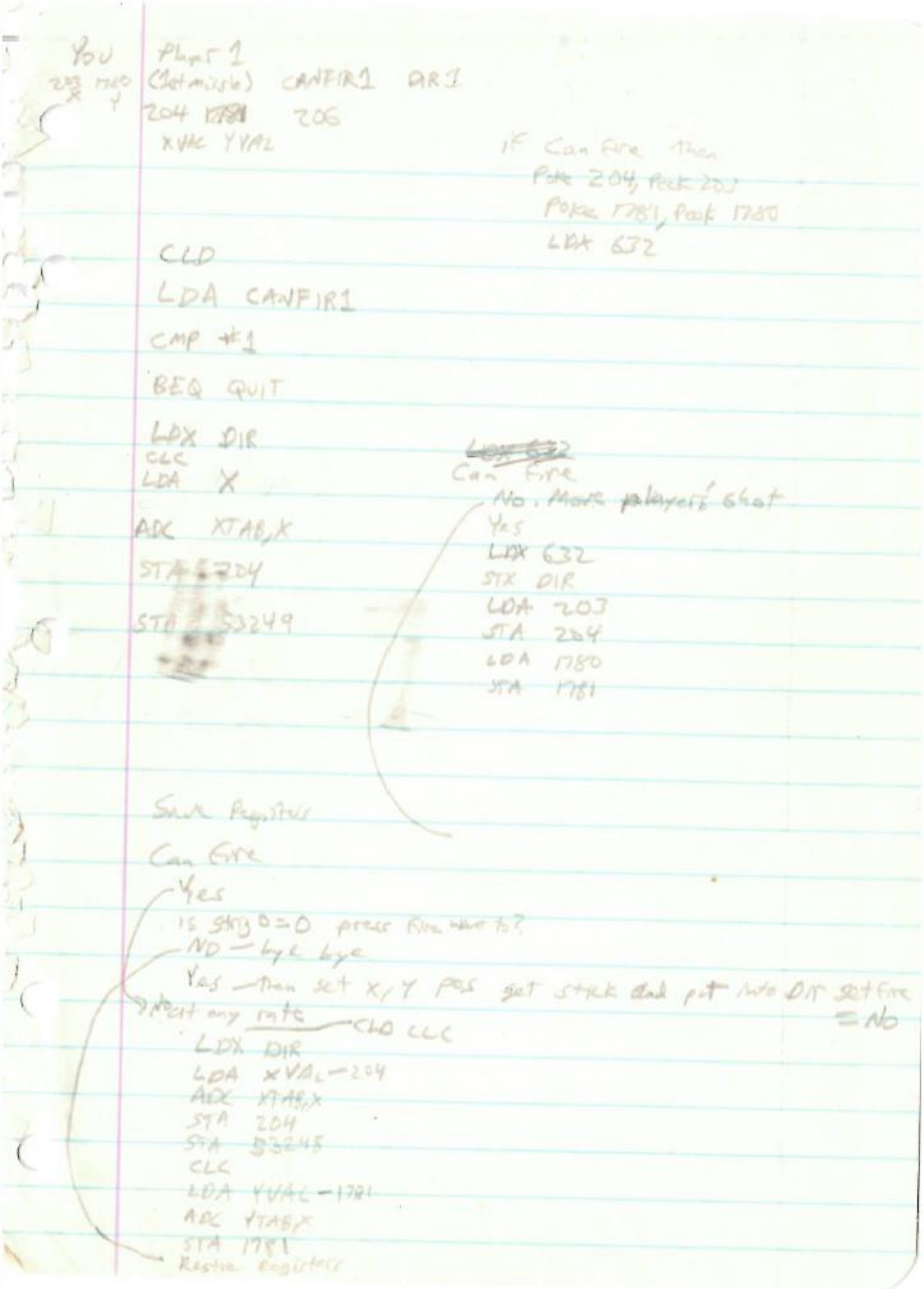
1536- 1548

72,138,72,14,174,120,2,136,133.

203,104,170,104,64

POKE 54286,192

?, Peek 103



You | Player 1
203 1780 (Get missile) CANFIR1 DIR1
X Y
204 1781 206
XVAL YVAL

If Can Fire then
Poke 204, Peek 203
Poke 1781, Peek 1780
LDA 632

```
CLD
LDA CANFIR1
CMP #1
BEQ QUIT
LDX DIR
CLC
LDA X
ADC XTAB,X
STA 204
STA 53249
```

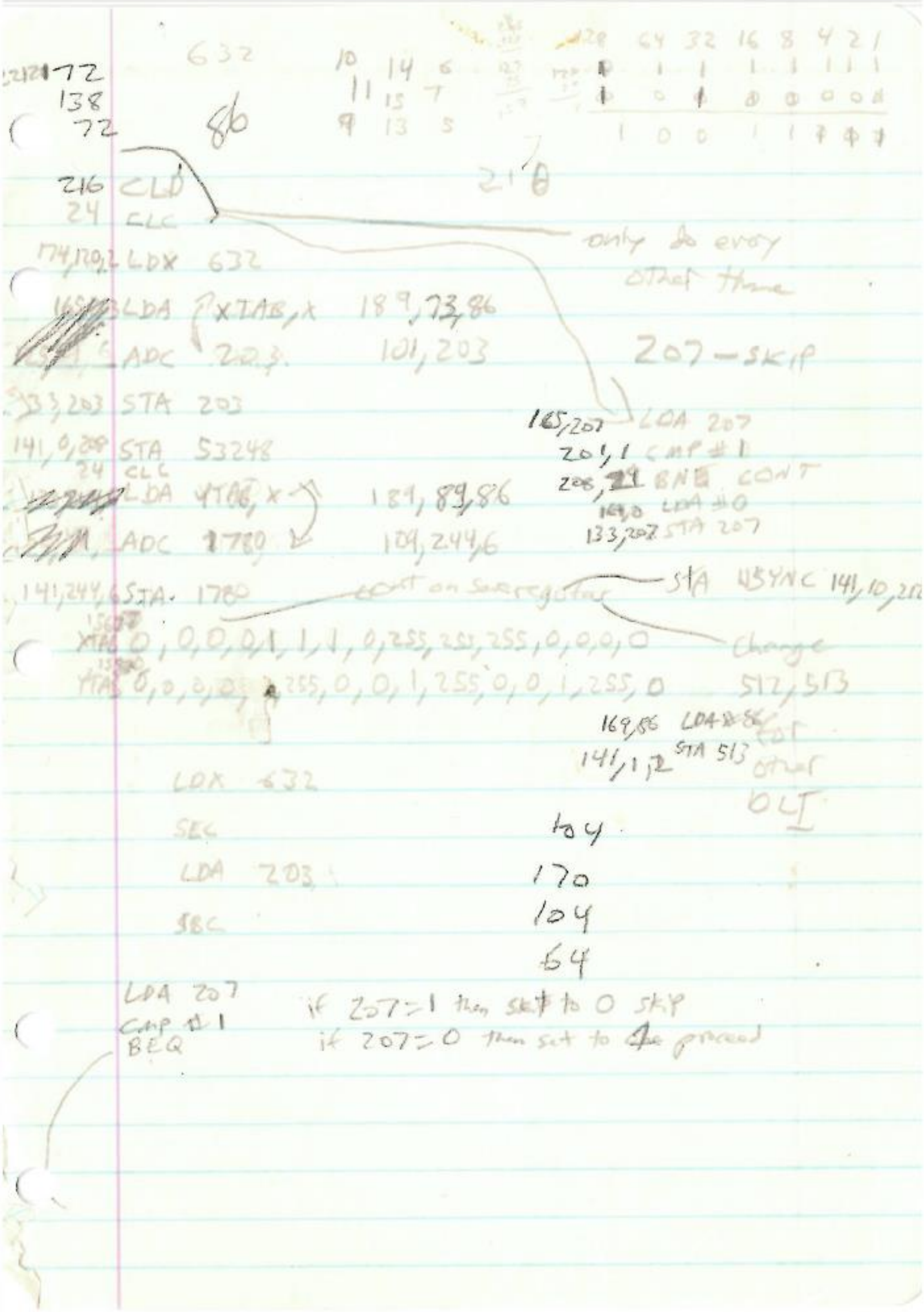
~~LDX 632~~
Can Fire
No. Move player's shot
Yes

```
LDX 632
STX DIR
LDA 203
STA 204
LDA 1780
STA 1781
```

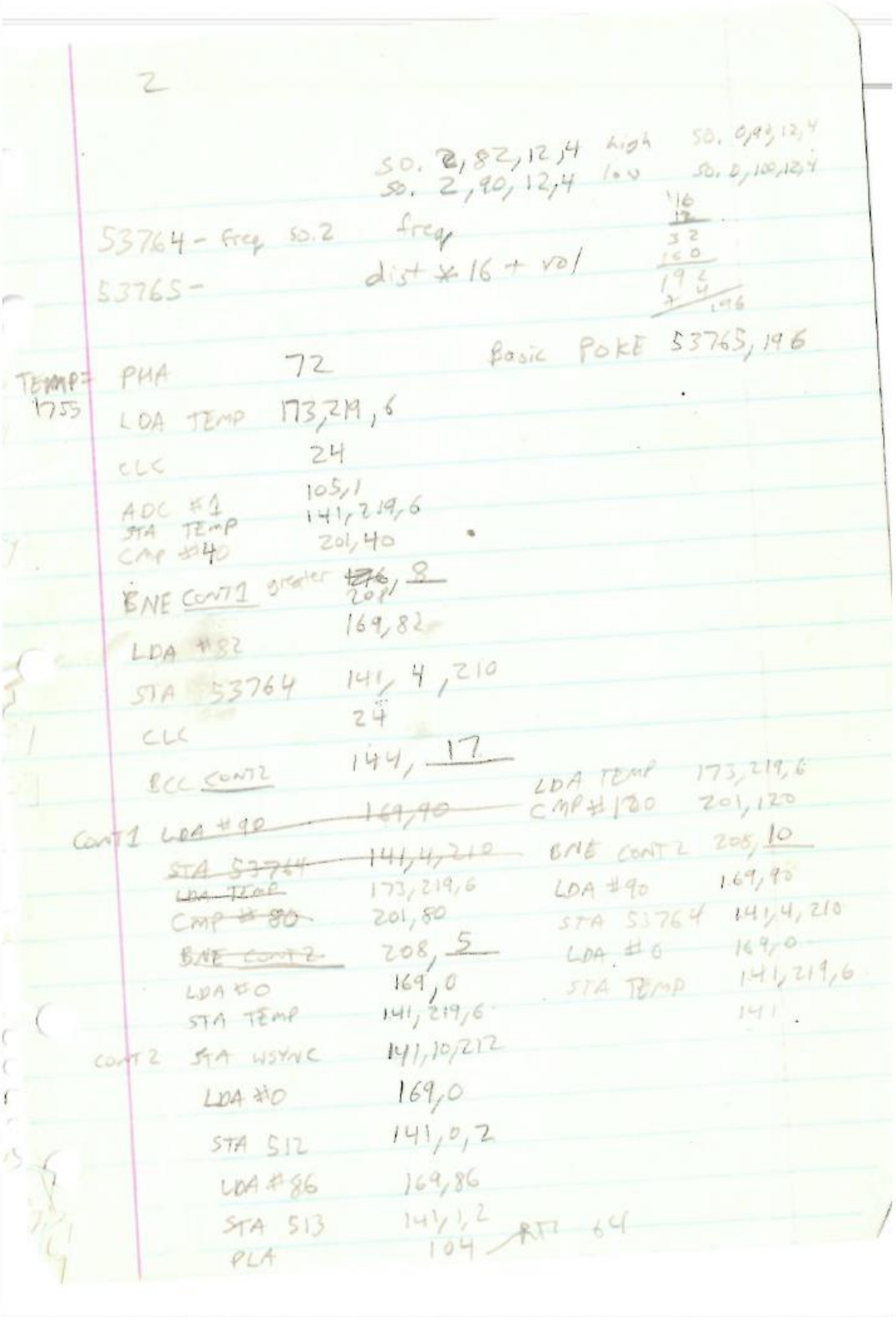
Save Registers
Can Fire
Yes
Is strig 0=0 press fire what to?
NO — bye bye
Yes — then set X/Y POS get stick and put into DIR set fire = No
Next any rate — CLD CLC

```
LDX DIR
LDA XVAL—204
ADC XTAB,X
STA 204
STA 53248
CLC
LDA YVAL—1781
ADC YTAB,X
STA 1781
```

Restore Registers



```
72        632         10  14  6        128 64 32 16 8 4 2 1
138                    11  15  7         1  1  1  1 1 1 1 1
72        86           9  13  5          1  0  1  0 0 0 0 0
                                         1  0  0  1 1 1 0 1
                                21'0
216       CLD
24        CLC                                  only do every
174,120,2 LDX   632                            other time
165,[illegible] LDA   XTAB,x     189,73,86
[illegible] ADC   203          101,203                    207 - skip
133,203   STA   203
141,0,208 STA   53248                          165,207 LDA 207
24        CLC                                  201,1   CMP #1
[illegible] LDA   YTAB,x     189,89,86             208,[illegible] BNE CONT
[illegible] ADC   1780         109,244,6               169,0   LDA #0
141,244,6 STA.  1780                           133,207 STA 207
                  cont on save register ——— STA WSYNC 141,10,212
15600
XTAB 0,0,0,0,1,1,1,0,255,255,255,0,0,0,0       change
15800
YTAB 0,0,0,0,255,0,0,1,255,0,0,1,255,0         512,513
                                               169,86   LDA #86    for
                                               141,1,2  STA 513    other
                                                                   DLI
     LDX  632
     SEC                        104
     LDA  203                   170
     SBC                        104
                                64
LDA 207      if 207=1 then set to 0 skip
CMP #1       if 207=0 then set to 1 proceed
BEQ
```



2

SO. 2,82,12,4 high SO. 0,90,12,4
SO. 2,90,12,4 low SO. 0,100,12,4

53764 - freq SO.2 freq

53765 - dist * 16 + vol

16
12
32
160
192
+ 4
196

Basic POKE 53765,196

TEMP= 1755

PHA 72
LDA TEMP 173,219,6
CLC 24
ADC #1 105,1
STA TEMP 141,219,6
CMP #40 201,40
BNE CONT1 greater ~~176~~ 208, 8
LDA #82 169,82
STA 53764 141, 4 ,210
CLC 24
BCC CONT2 144, 17

CONT1 ~~LDA #90~~ ~~169,90~~
~~STA 53764~~ ~~141,4,210~~
~~LDA TEMP~~ 173,219,6
~~CMP #80~~ 201,80
~~BNE CONT2~~ 208, 5
LDA #0 169,0
STA TEMP 141,219,6
CONT2 STA WSYNC 141,10,212
LDA #0 169,0
STA 512 141,0,2
LDA #86 169,86
STA 513 141,1,2
PLA 104 RTI 64

LDA TEMP 173,219,6
CMP #120 201,120
BNE CONT2 208, 10
LDA #90 169,90
STA 53764 141,4,210
LDA #0 169,0
STA TEMP 141,219,6
141 .

1752 — returns value   1753   1754
1 - if hit monster   X coord   Y coord
0 - it wall   m/s   m/s

1755 0-3   3RD

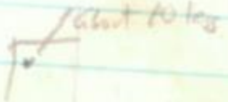
```
INC 1755
LDA 1755
CMP #4
BNE CONT0
LDA #0
STA 1755
CLC
BCC CONT1
```

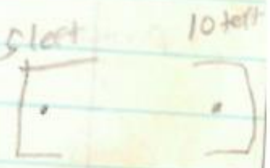
```
                     512  513
~~PLA~~              173, 86
~~PLA~~
                     22189
PHA                  72
CONT0 LDA 53253      173,5,208
CMP #0               201,0
BEQ CONT1            240, ~~48~~ 39
CMP #1               201,1 — 201,4
BNE CONT2            208, ~~15~~ 10 — BEQ — 240,   22200
LDA #1               169,0
~~STA 209~~          ~~133,209~~
STA 1781             ~~141,245,6~~ → 141,1,208
STA 207              133,207
~~STA 1752~~         ~~141,216,6~~
CLC
BCC CONT4            144, ~~21~~ 25   24
CONT2 CMP #4         201,4 — 4
BNE CONT4            208, ~~25~~ 21
LDA 209              165,209
STA 1753             141,217,6
LDA 1781             173,245,6
STA 1754             141,218,6
LDA #1               169,0
~~STA 209~~          ~~133,209~~
STA 1781             ~~141,245,6~~ → 141,1,208
STA 207              133,207 — 209
LDA #1               169,1   Can GTE, no
STA 1752             141,216,6
CONT4 LDA #0         ~~169,0~~
STA 53278            ~~141,30,208~~
```

```
238,219,6
173,219,6
201,4
208,8
169,0
141,219,6
24
144,[illegible]
```

```
CONT1 STA WSYNC   141,10,212
LDA #0            169,0
STA 513           141,0,2
PLA               104
RTI               64    22259
                  2 23
```

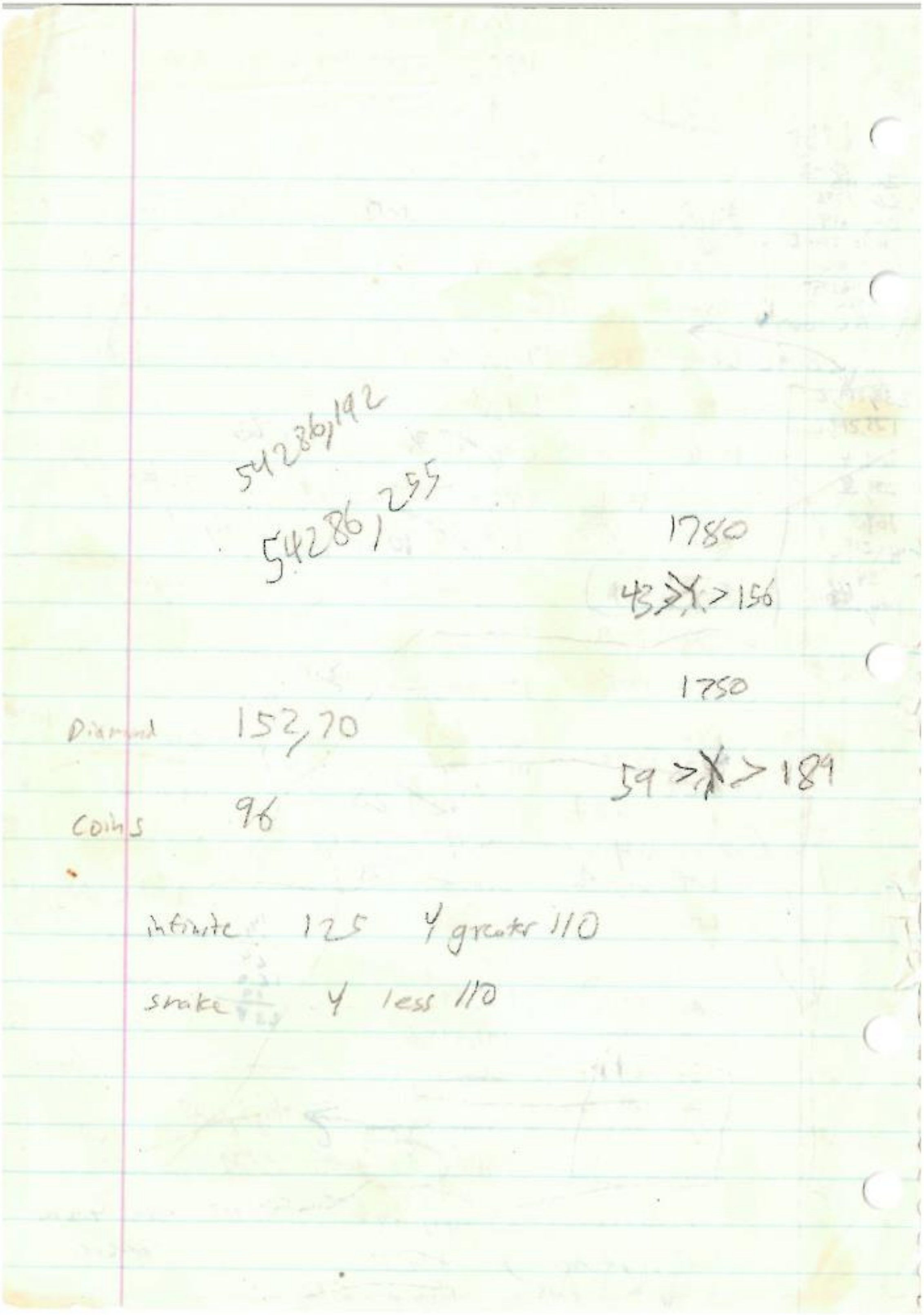
30 less for both Y





16
14 ×14
64
160
14
238

Poke [illegible],0
Then
CLR,0
and restore
and
and change
and
Set yes in
basic



54286,192

54286,255

1780

43 > X > 156

1750

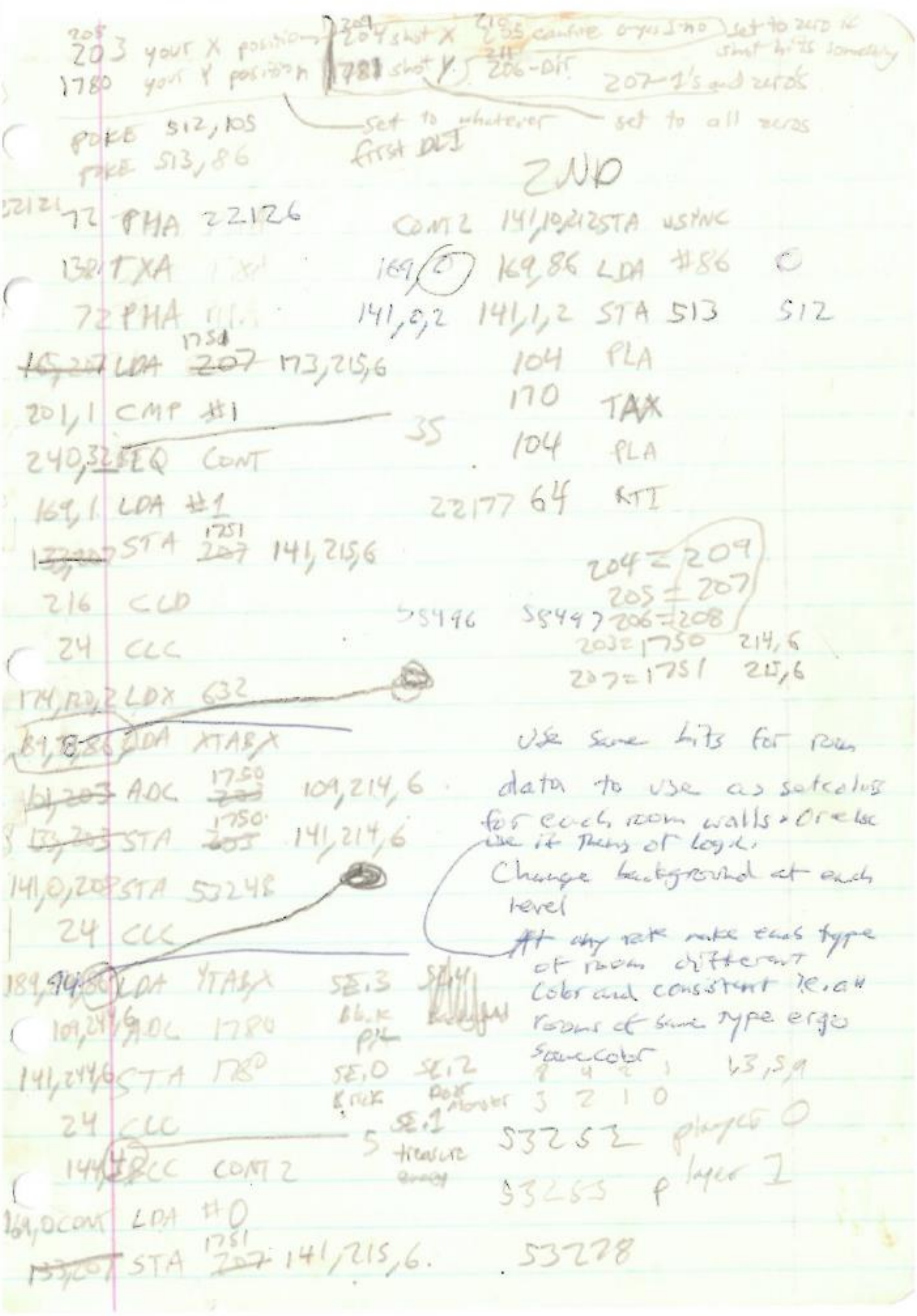
Diamond 152,70

59 > X > 189

Coins 96

infinite 125 Y greatr 110

snake Y less 110



205
203 your X position 209 204 shot X 210 35 confire 0-yes 1-no set to zero if shot hits something
1780 your Y position 1781 shot Y 211 206-Dif 207-1's and zeros
set to whatever first DLI set to all zeros

```
POKE 512,105
POKE 513,86
```

ZND

```
22121 72 PHA 22126
138 TXA
72 PHA
~~16,207~~ LDA 1750 ~~207~~ 173,215,6
201,1 CMP #1
240,32 BEQ CONT
169,1 LDA #1
~~133,207~~ STA 1751 ~~207~~ 141,215,6
216 CLD
24 CLC
174,120,2 LDX 632
189,8,86 LDA XTAB,X
~~101,203~~ ADC 1750 ~~203~~ 109,214,6
~~133,203~~ STA 1750 ~~205~~ 141,214,6
141,0,208 STA 53248
24 CLC
189,94,86 LDA YTAB,X
109,244,6 ADC 1780
141,244,6 STA 1780
24 CLC
144,4 BCC CONT2
169,0 CONT LDA #0
~~133,207~~ STA 1751 ~~207~~ 141,215,6.
```

35

```
CONT2 141,10,212 STA WSYNC
169,0 169,86 LDA #86 0
141,0,2 141,1,2 STA 513 512
104 PLA
170 TAX
104 PLA
22177 64 RTI
```

204 = 209
205 = 207
55496 55497 206 = 208
203 = 1750 214,6
207 = 1751 215,6

Use same bits for room data to use as setcolors for each room walls + or else use it things of logic. Change background at each level
At any rate make each type of room different color and consistent i.e. all rooms of same type ergo same color

SE.3 BLK P/L SE.4
SE.0 Brick SE.2 Box Monster
5 SE.1 treasure energy

8 4 2 1 1,3,5,9
3 2 1 0

53252 player 0
53253 player 1
53278

```
206 DIR
205 CANFIR1
    1 = No
    0 = Yes
```

In basic, if player one hits something Poke 204, 1781, CANFIR to zero

```
22016
72        PHA
138       TXA
72        PHA
165,205   LDA CANFIR1
201,1     CMP #1          34
240,33    BEQ CONT1
173,132,2 LDA 644
201,1     CMP #1          51
240,50    BEQ CONT 2
173,120,2, LDA 632        CMP #15   201,15
                          BEQ CONT 2   240,43
165,203   LDA 203
133,204   STA 204         add
173,244,6 LDA 1780
141,245,6 STA 1781
173,120,2 LDA 632
133,206   STA DIR
169,1     LDA #1
133,205   STA CANFIR1
216       CLD
24 CONT1  CLC
166,206   LDX DIR
165,204   LDA 204
125,78 86 ADC XTAB,X
133,204   STA 204
141,1,208 STA 53249
24        CLC
```

```
FIRST 173,245,6  LDA 1781
      125,94 86  ADC YTAB,X
      141,245,6  STA 1781
      141,10,212 CONT2 STA WSYNC
      104        PLA
      170        TAX
      104        PLA
      64         RTI
22088
```

```
169,125,141,0,2
LDA #105
STA 512
```

```
22089 XTAB                73
  0,0,0,0,0,1,1,1,0,255,255,255,0,0,0,0
22105
YTAB
  0,0,0,0,0,1,255,0,0,1,255,0,0,1,
  255,0 22120
```

Your X 1750

Your Y 1780

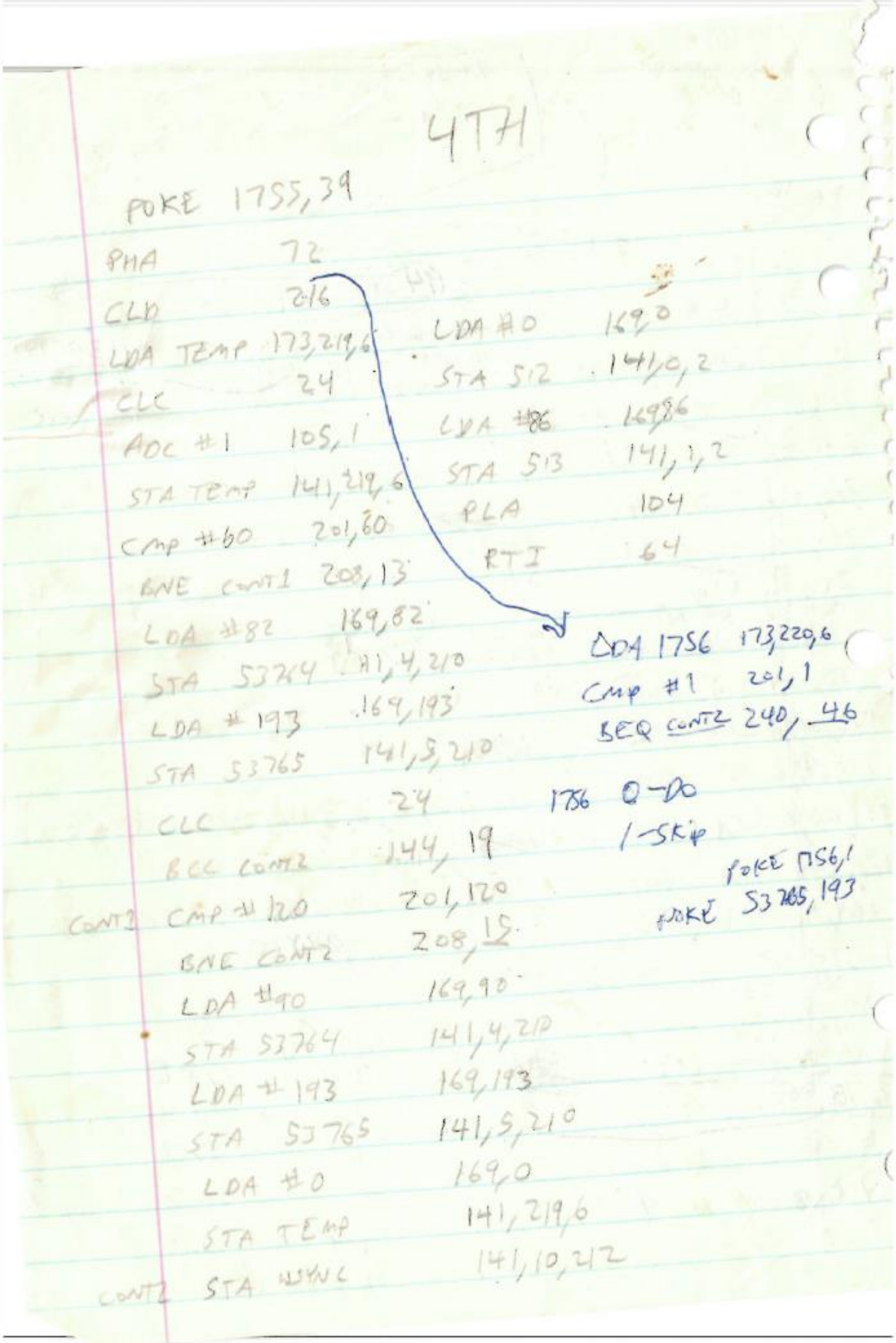
Shot X = 204

Shot y = 1781

CanFir = 207 0 Yes 1 No

Dir = 208

Skip = 1781



4TH

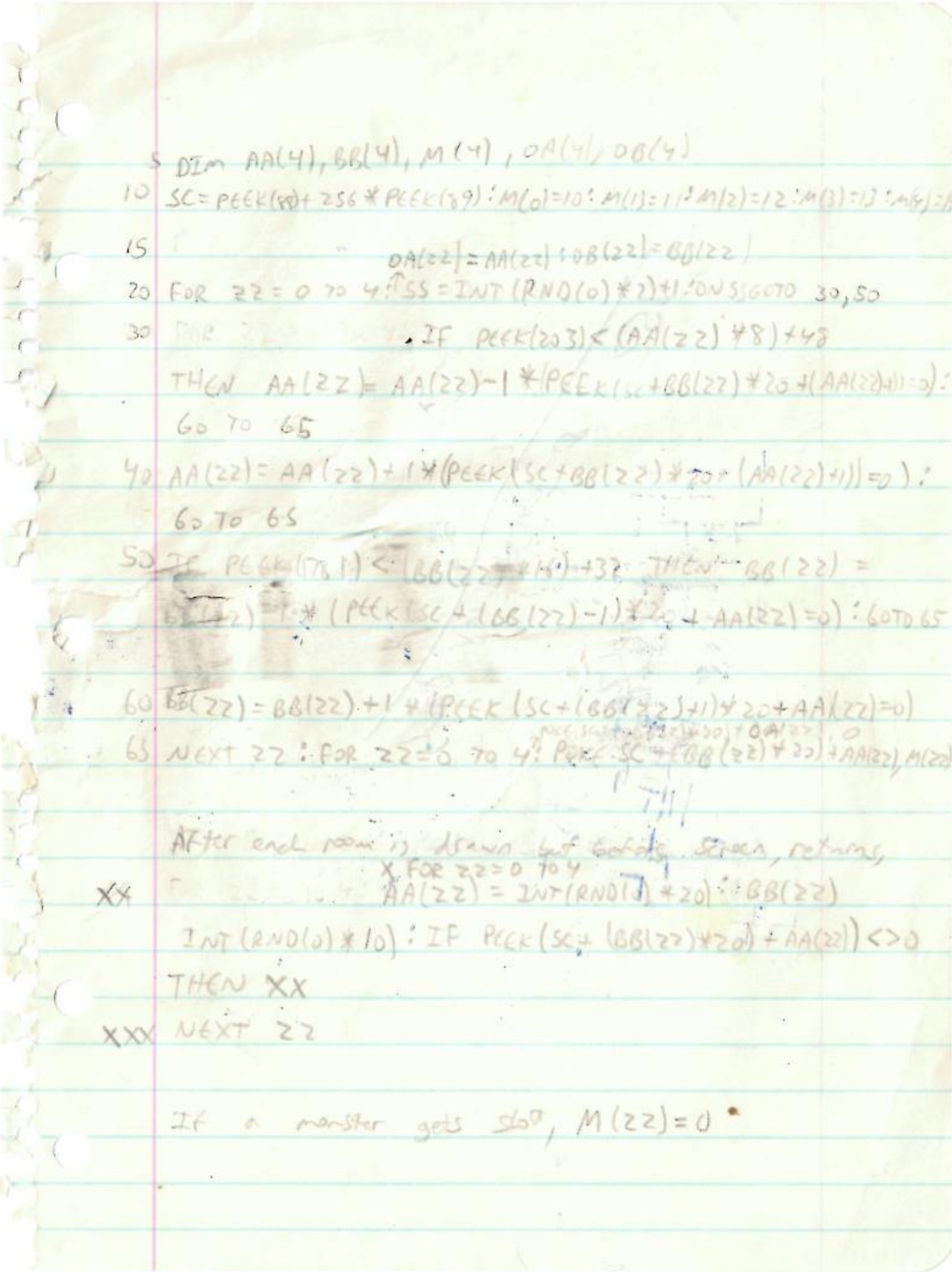
```
POKE 1755,39
PHA              72
CLD              216
LDA TEMP         173,219,6
CLC              24
ADC #1           105,1
STA TEMP         141,219,6
CMP #60          201,60
BNE CONT1        208,13
LDA #82          169,82
STA 53764        141,4,210
LDA #193         169,193
STA 53765        141,5,210
CLC              24
BCC CONT2        144,19
CONT1 CMP #120   201,120
BNE CONT2        208,15
LDA #90          169,90
STA 53764        141,4,210
LDA #193         169,193
STA 53765        141,5,210
LDA #0           169,0
STA TEMP         141,219,6
CONT2 STA WSYNC  141,10,212
```

```
LDA #0     169,0
STA 512    141,0,2
LDA #86    169,86
STA 513    141,1,2
PLA        104
RTI        64
```

```
LDA 1756    173,220,6
CMP #1      201,1
BEQ CONT2   240,46
```

1756 0-DO
1-SKip

POKE 1756,1
POKE 53765,193



```
5 DIM AA(4),BB(4),M(4),OA(4),OB(4)
10 SC=PEEK(88)+256*PEEK(89):M(0)=10:M(1)=11:M(2)=12:M(3)=13:M(4)=1
15                 OA(ZZ)=AA(ZZ):OB(ZZ)=BB(ZZ)
20 FOR ZZ=0 TO 4:SS=INT(RND(0)*2)+1:ON SS GOTO 30,50
30 FOR              .IF PEEK(203)<(AA(ZZ)*8)+48
   THEN AA(ZZ)=AA(ZZ)-1*(PEEK(SC+BB(ZZ)*20+(AA(ZZ)-1))=0):
   GO TO 65
40 AA(ZZ)=AA(ZZ)+1*(PEEK(SC+BB(ZZ)*20+(AA(ZZ)+1))=0):
   GO TO 65
50 IF PEEK(781)<(BB(ZZ)*8)+32 THEN BB(ZZ)=
   BB(ZZ)-1*(PEEK(SC+(BB(ZZ)-1)*20+AA(ZZ)=0):GOTO 65

60 BB(ZZ)=BB(ZZ)+1*(PEEK(SC+(BB(ZZ)+1)*20+AA(ZZ)=0)
                                 POKE SC+(OB(ZZ)*20)+OA(ZZ),0
65 NEXT ZZ:FOR ZZ=0 TO 4:POKE SC+(BB(ZZ)*20)+AA(ZZ),M(ZZ)
```

After each room is drawn but before screen returns,

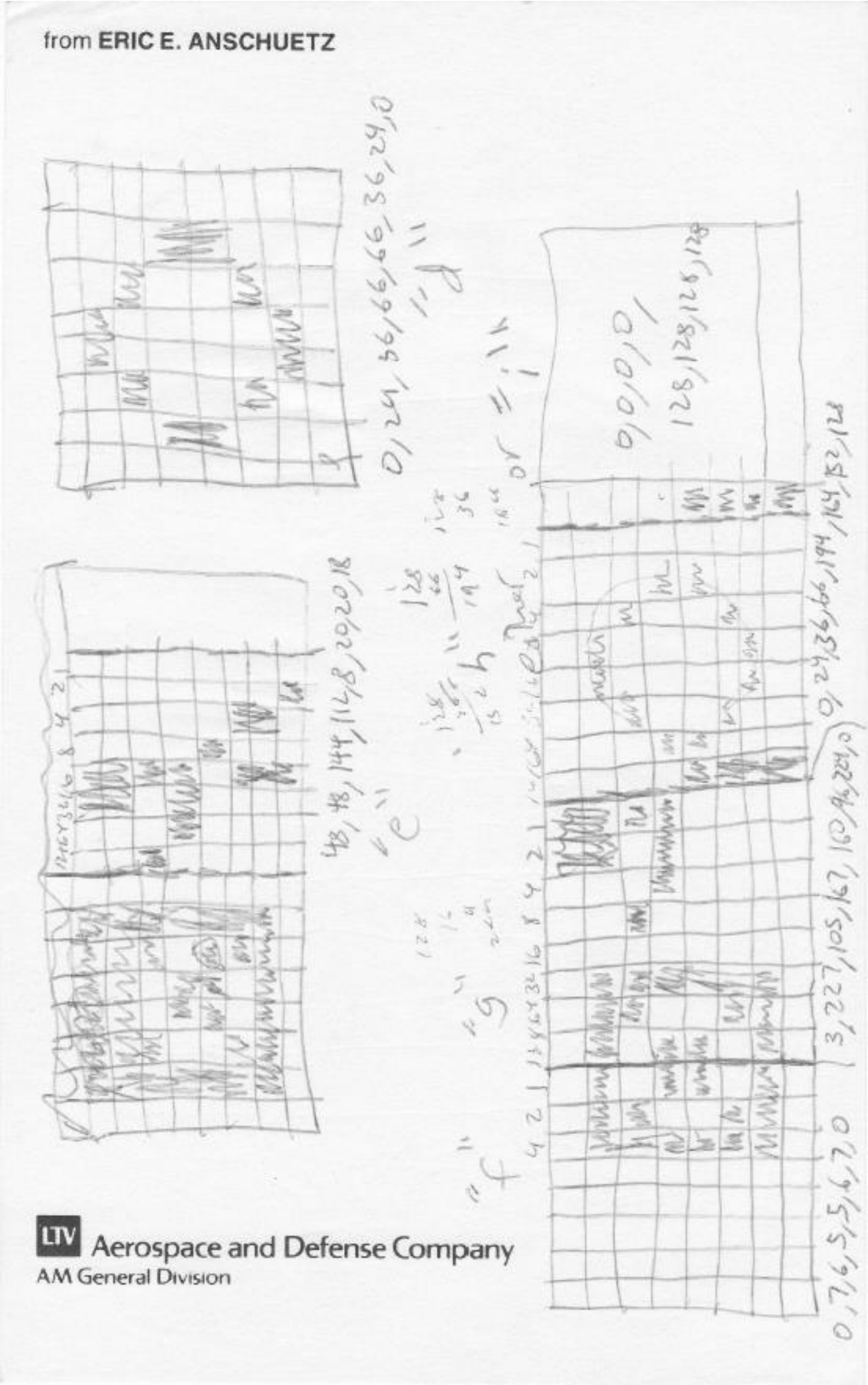
```
              X FOR ZZ=0 TO 4
XX                AA(ZZ)=INT(RND(0)*20):BB(ZZ)
   INT(RND(0)*10):IF PEEK(SC+(BB(ZZ)*20)+AA(ZZ))<>0
   THEN XX
XXX NEXT ZZ
```

If a monster gets shot, M(ZZ)=0

182, 147

Stop interrupt at. 250

SOKO-BAN



from ERIC E. ANSCHUETZ

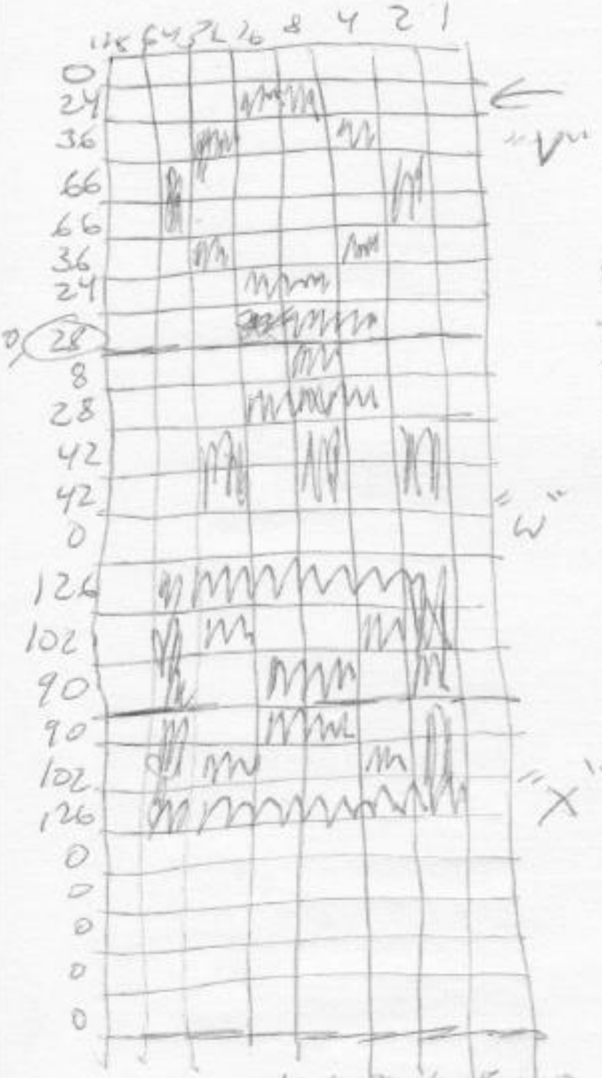
0,24,36,66,66,36,24,0
= "f"
or = "i"
0,0,0,0
128,128,128,128
48,48,144,112,8,20,20,18
= "e"
= "h"
128 64 32 16 8 4 2 1
0,24,36,66,194,194,152,128
3,227,105,167,101,96,224,0
= "g"
= "f"
0,7,6,5,5,6,7,0

LTV Aerospace and Defense Company
AM General Division

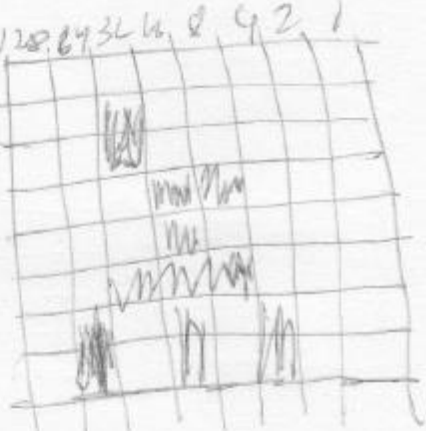
from **ERIC E. ANSCHUETZ**

"u" ~~0,0,0,0,~~

0,0,0,0,40,0,28

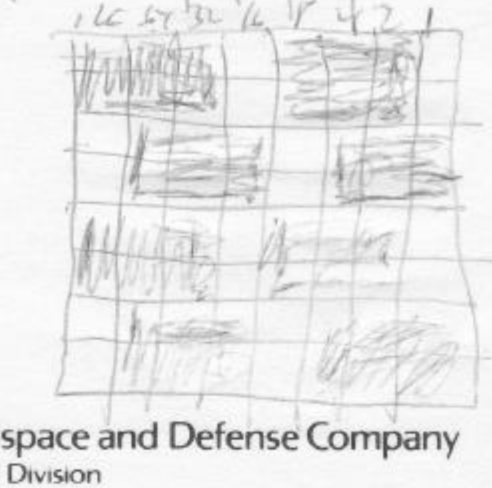


"t"



0,32,32,24,16,
56,86,86

255
17
238

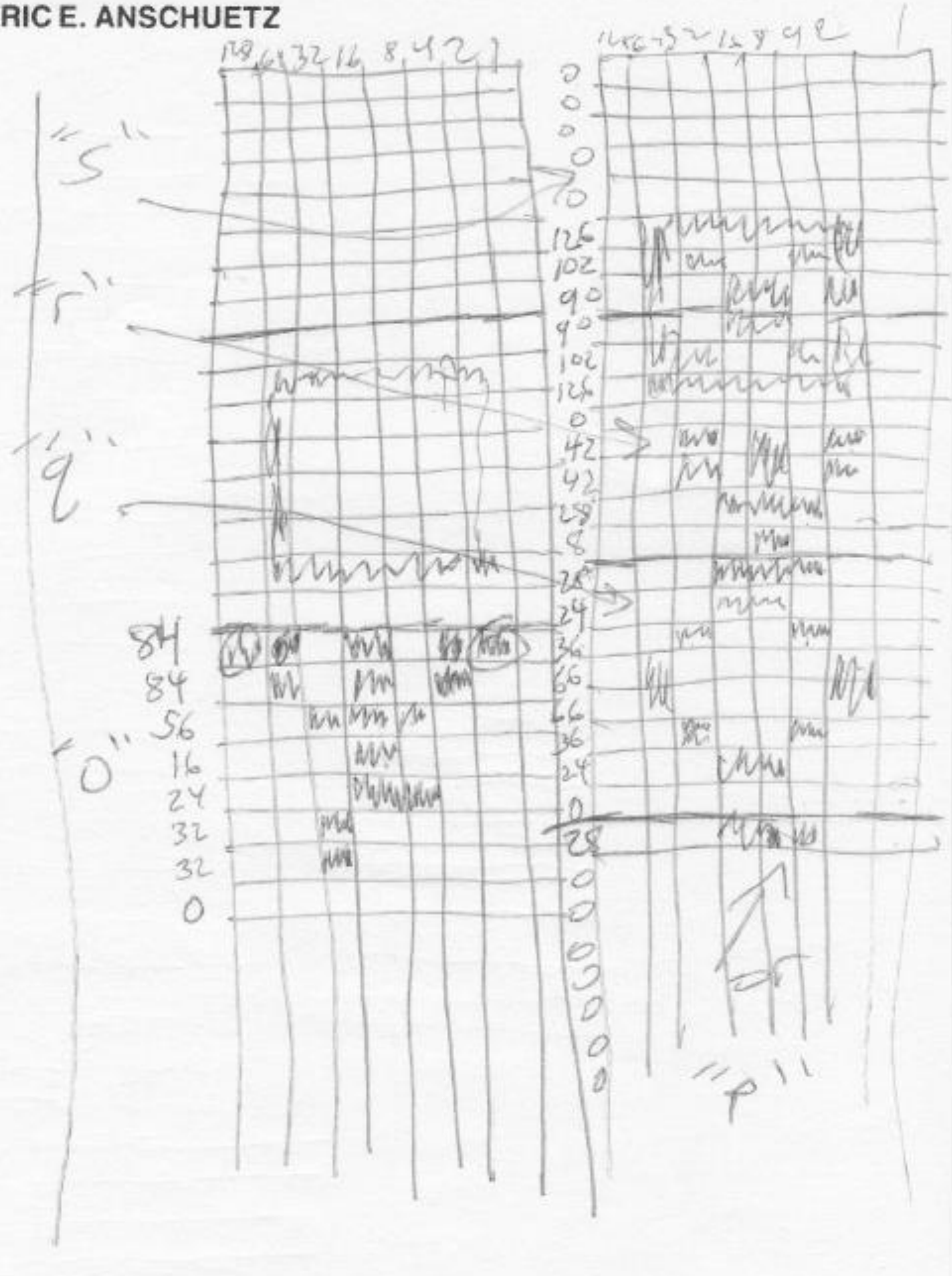


238,238,119,119,
238,238,119,119

"b"

LTV Aerospace and Defense Company
AM General Division

from **ERIC E. ANSCHUETZ**



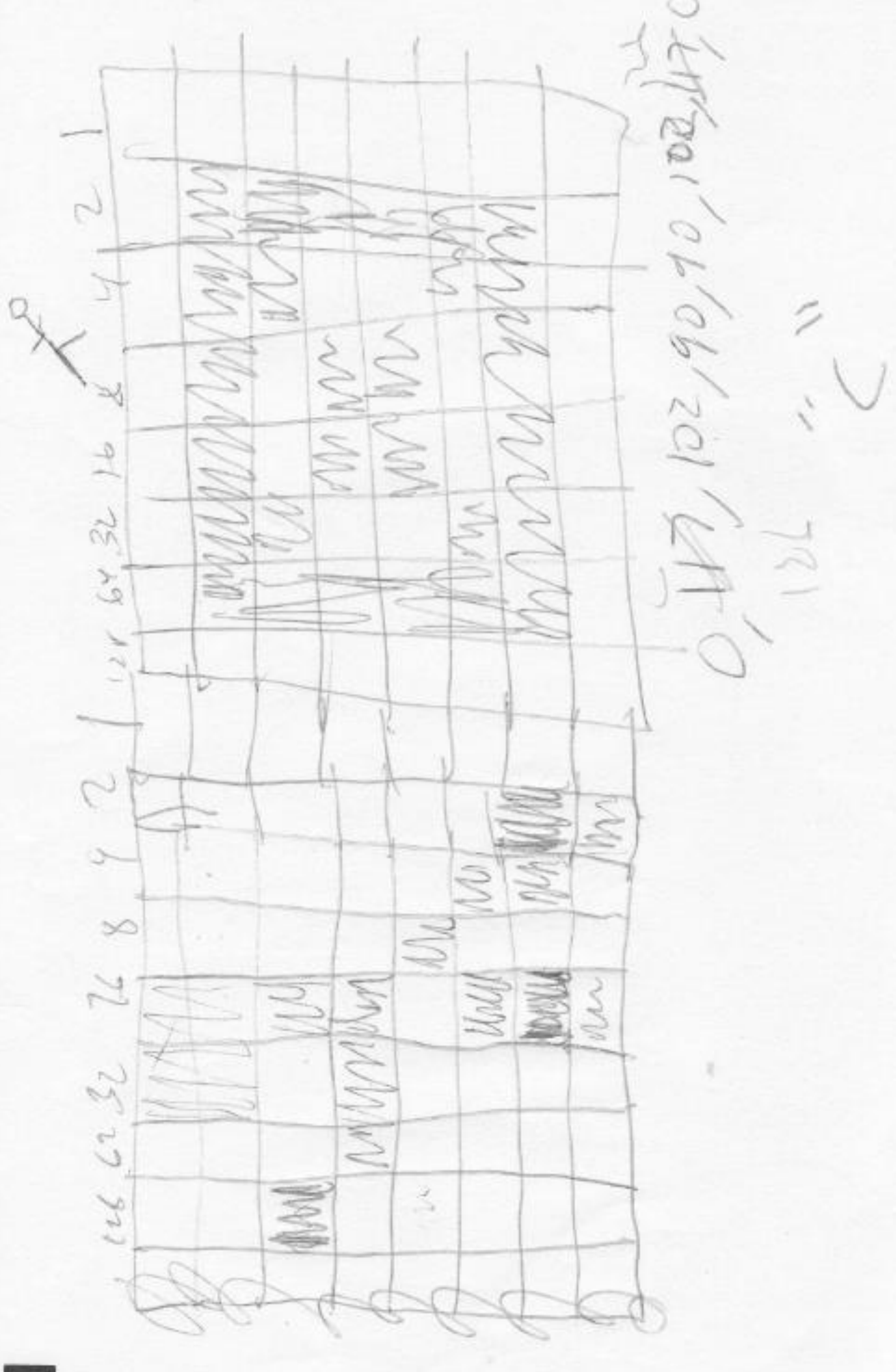
LTV Aerospace and Defense Company
AM General Division

from **ERIC E. ANSCHUETZ**

24/88/48/30/6/6/40/36

LTV Aerospace and Defense Company
AM General Division

from **ERIC E. ANSCHUETZ**



LTV Aerospace and Defense Company
AM General Division

from **ERIC E. ANSCHUETZ**

14 700

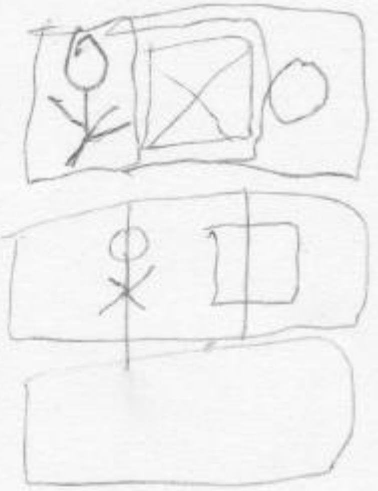
11 ← 15 → 7 600

500 13

600

```
  20
  16
 ----
 120
 200
 ----
 320
```

```
SCREEN1$(1,20) = "                    "
 SCREEN1$(21,40) = "                    "
 SCREEN1$(41,80) = "                    "
 SCREEN1$(81,80) = "□□□12W□□"
```

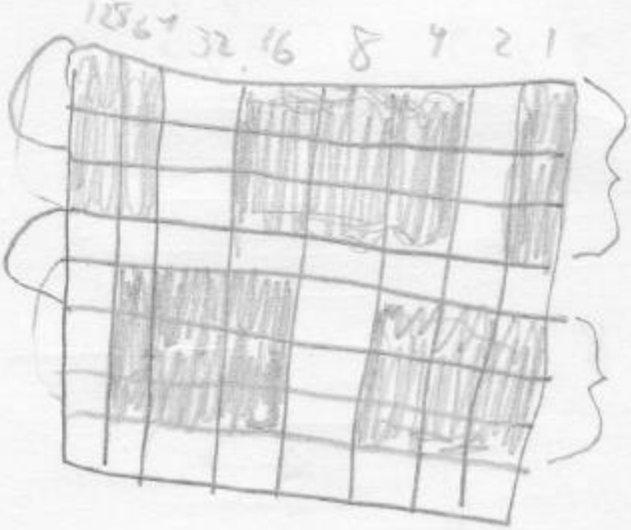


LTV Aerospace and Defense Company
AM General Division

from **ERIC E. ANSCHUETZ**

1 2
128
64
29

221



1
64
0 48
7
119

LTV Aerospace and Defense Company
AM General Division

1

1 2 2

1 2

1 2 2 2

4 4

3 3 3 4 4 4

3 3 4

3 3 3 ~~4~~

~~3 3 3~~

5 5 5

5

5 8 8 8

6

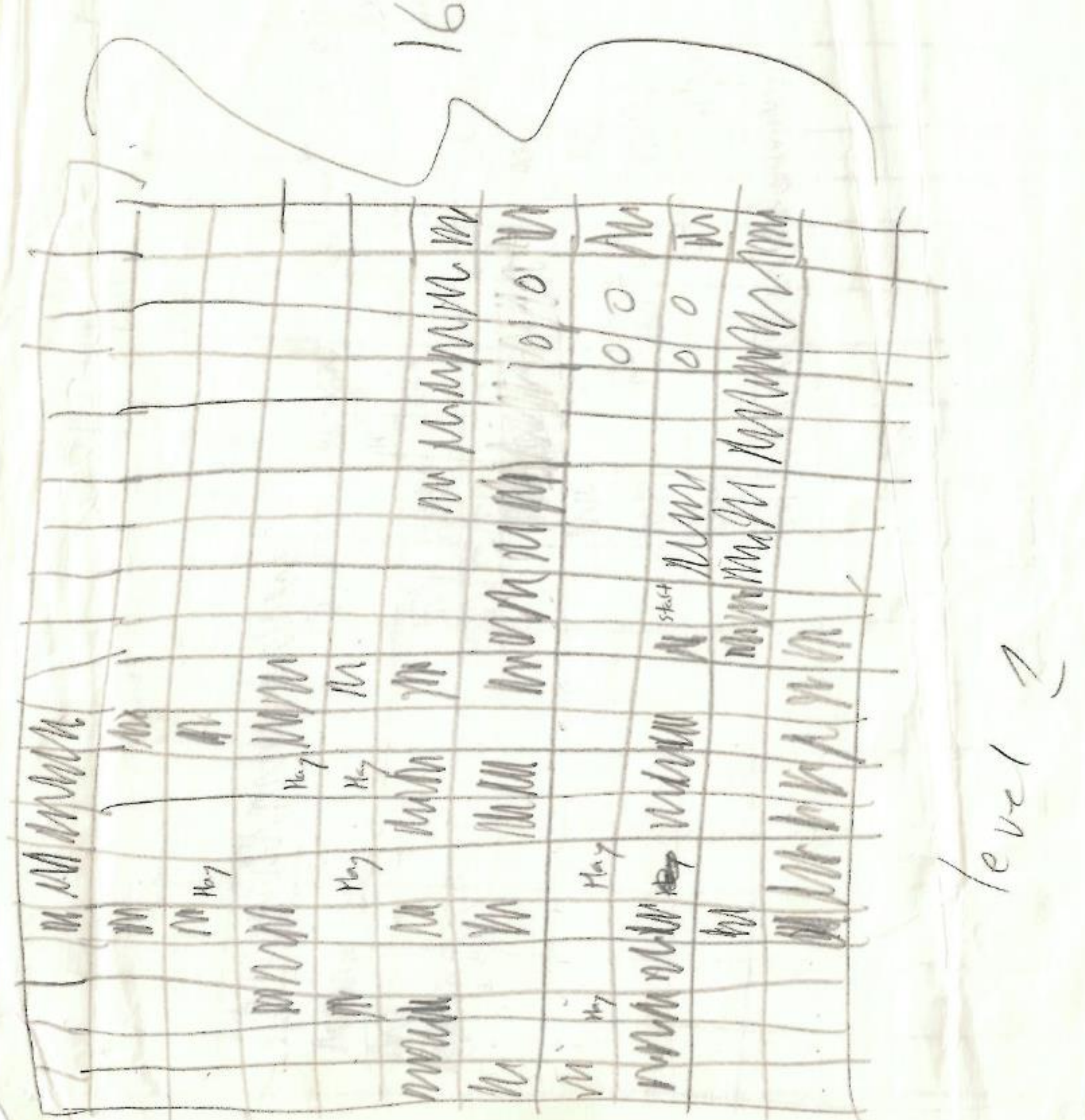
6 6 6 6 6 6 7 7 7

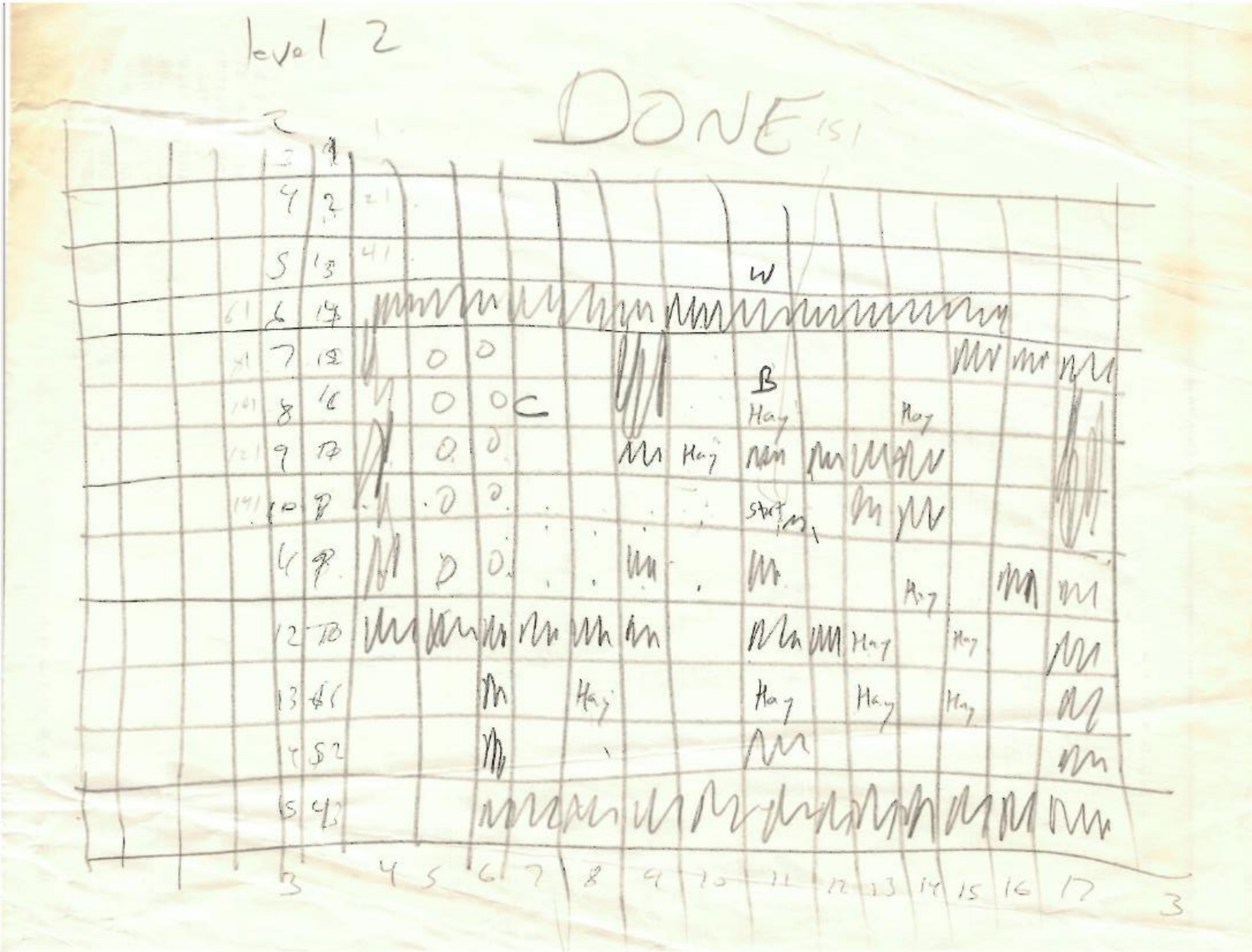
6 6 7

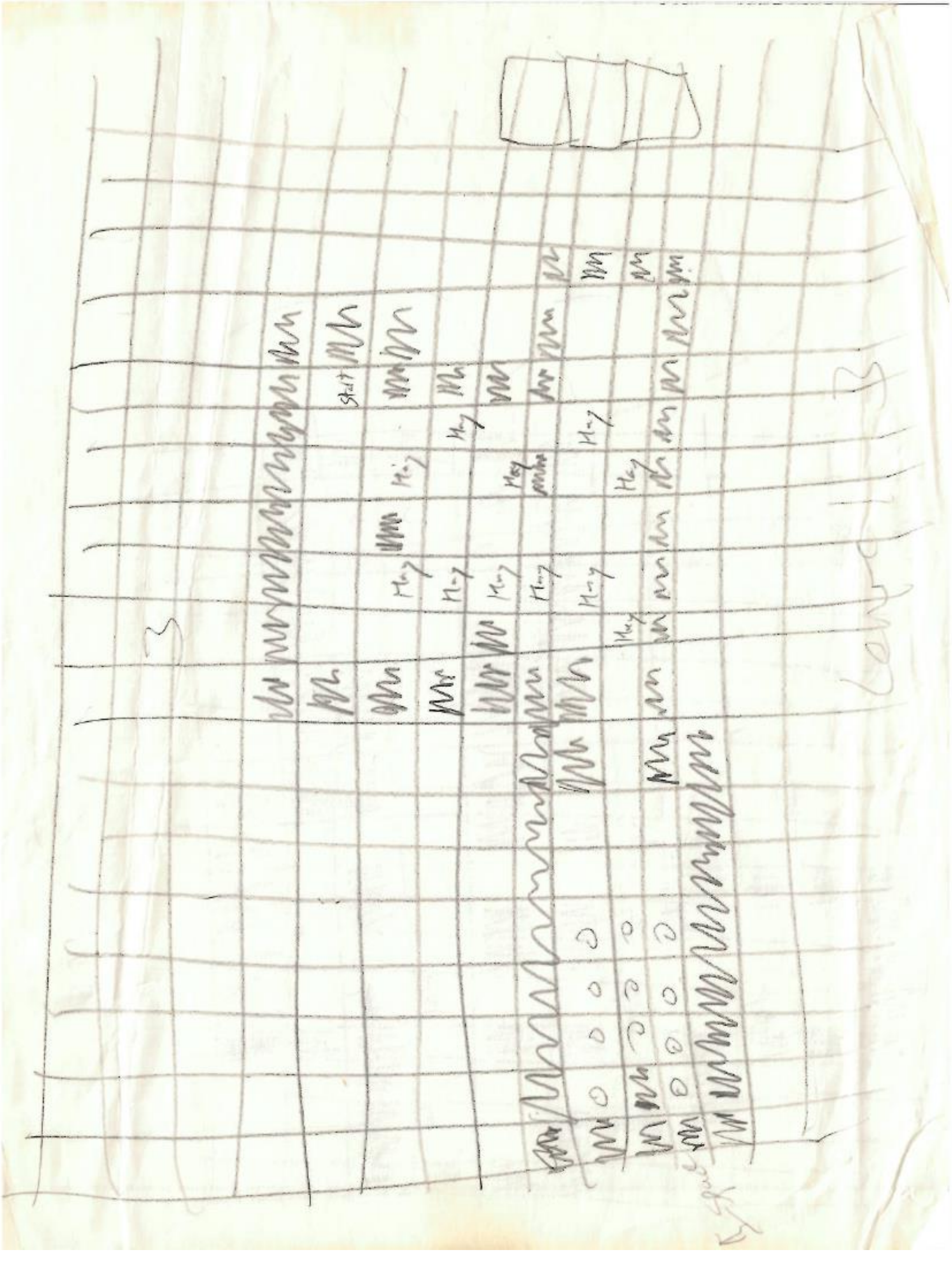
6 6 6 6 7

6 6 6 6 6 7

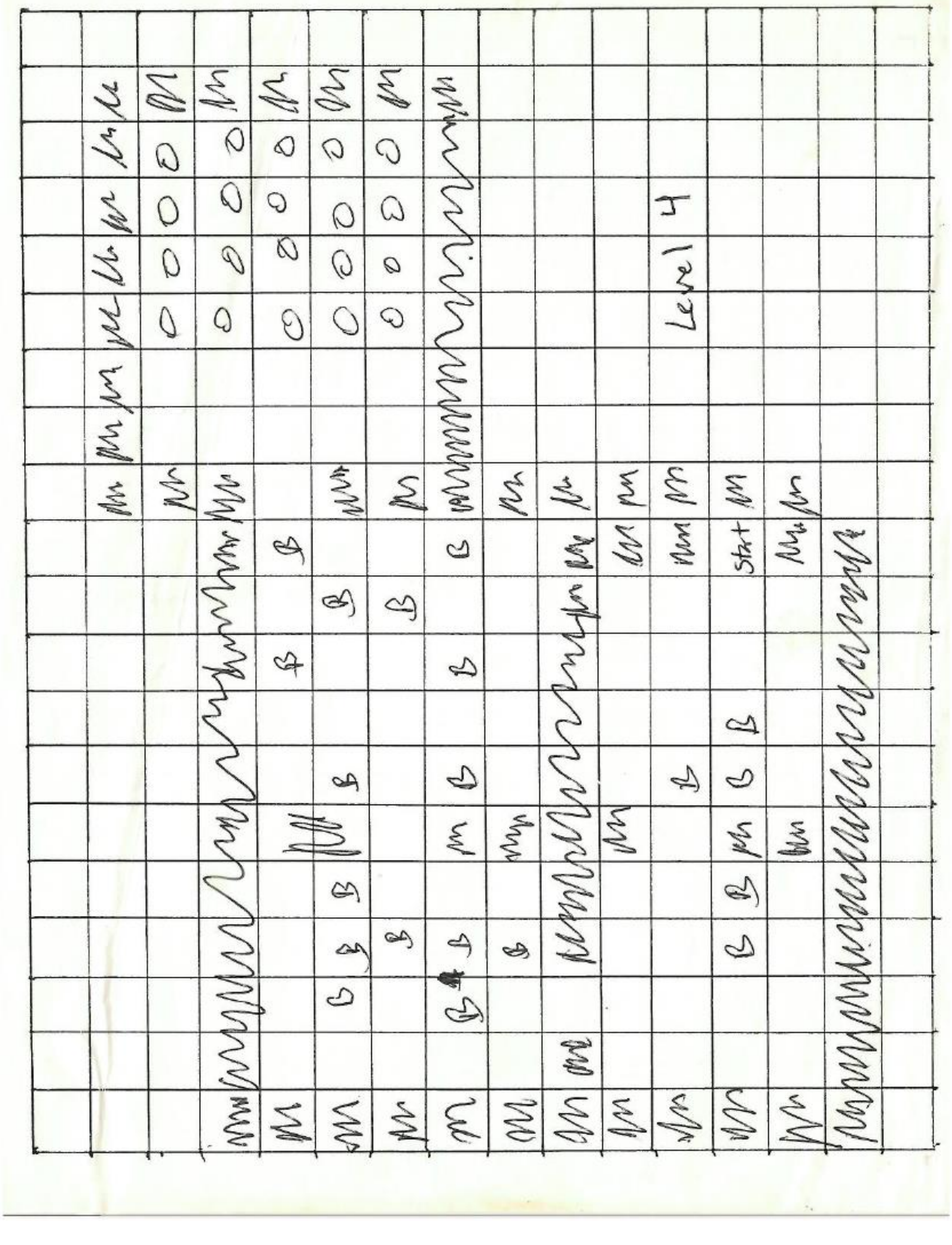
6

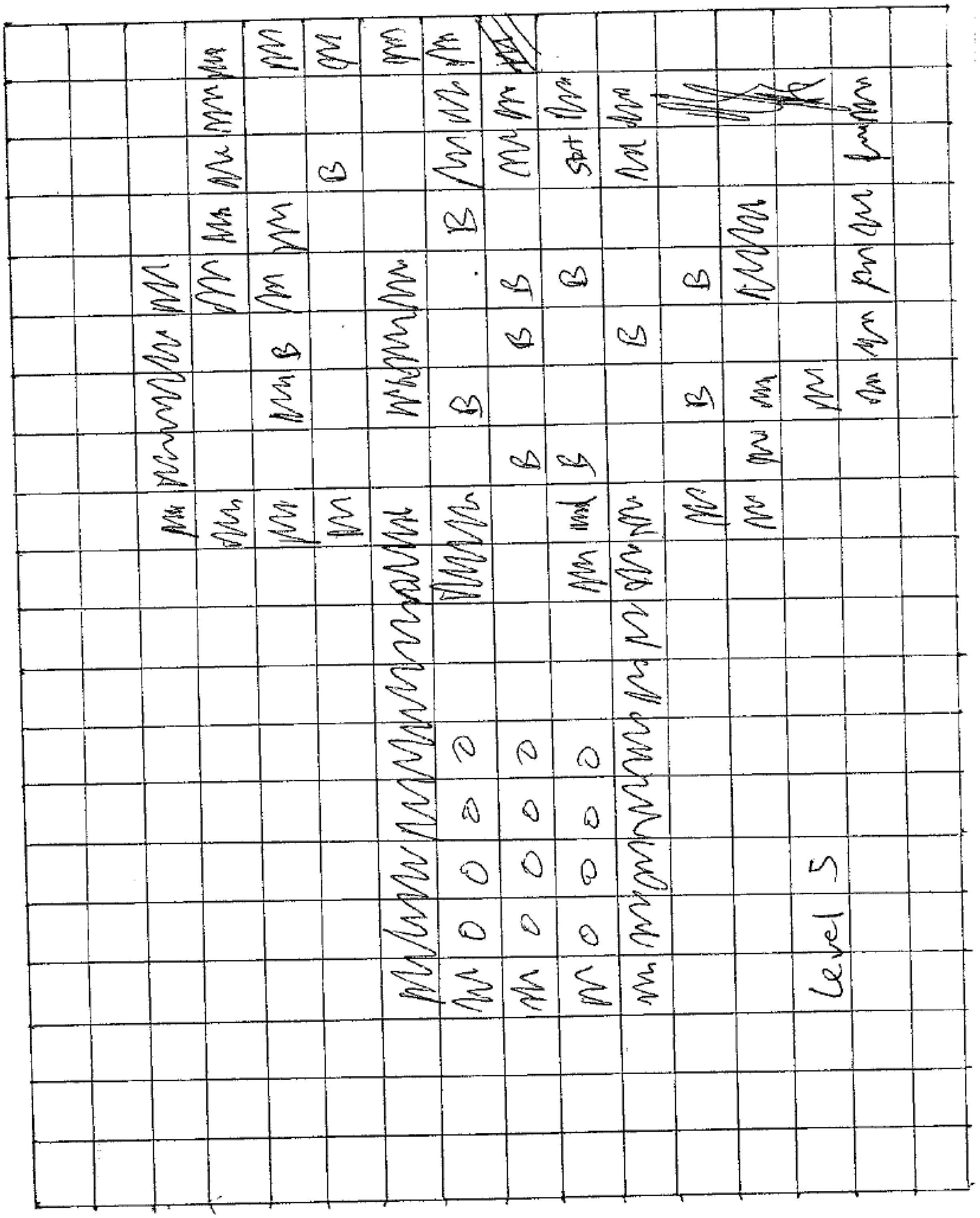

16
Hay
Hay
Hay
Hay
Hay
Hay
Start
level 2

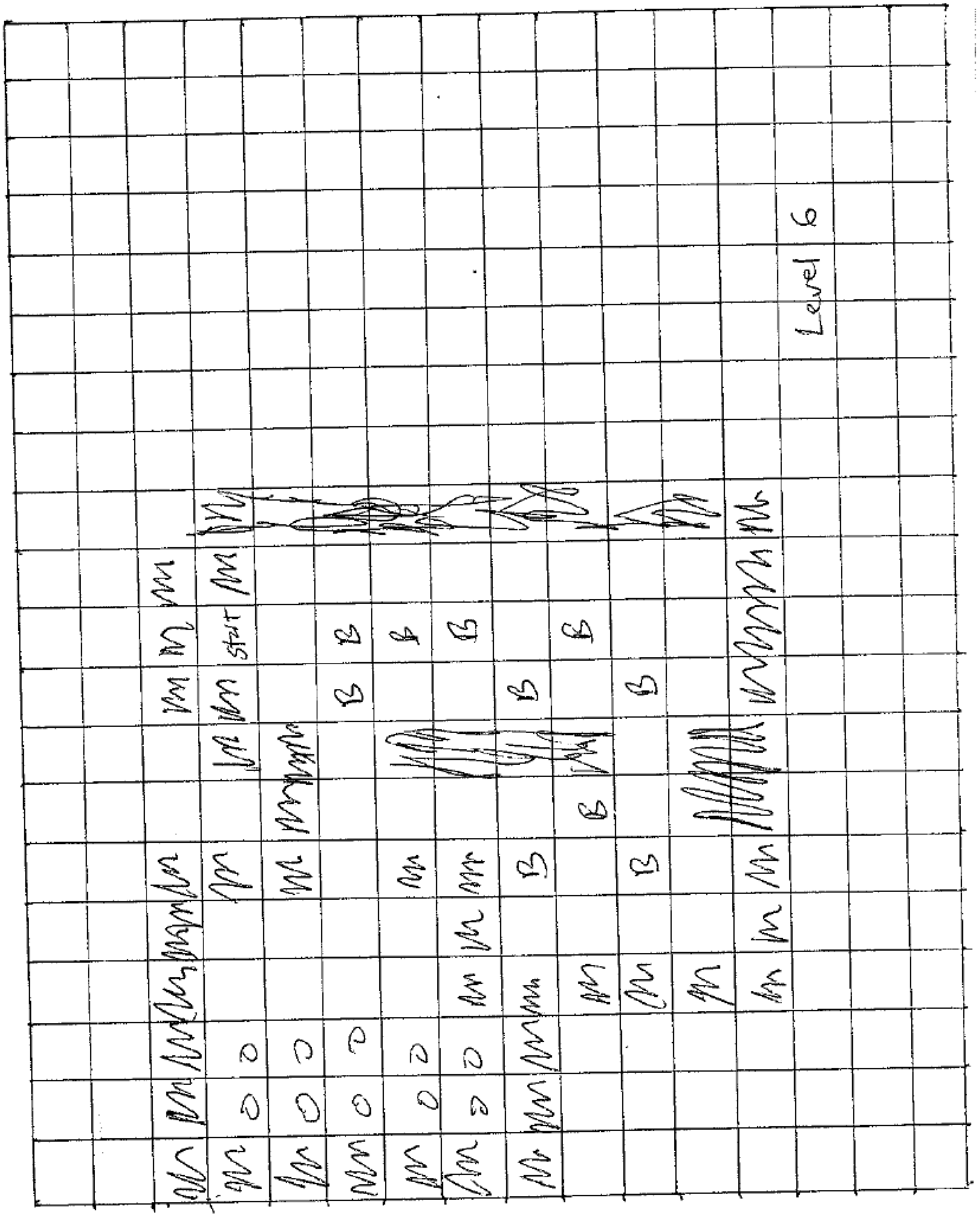

level 2
DONE 151
W
B
Hay
C
start
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
3

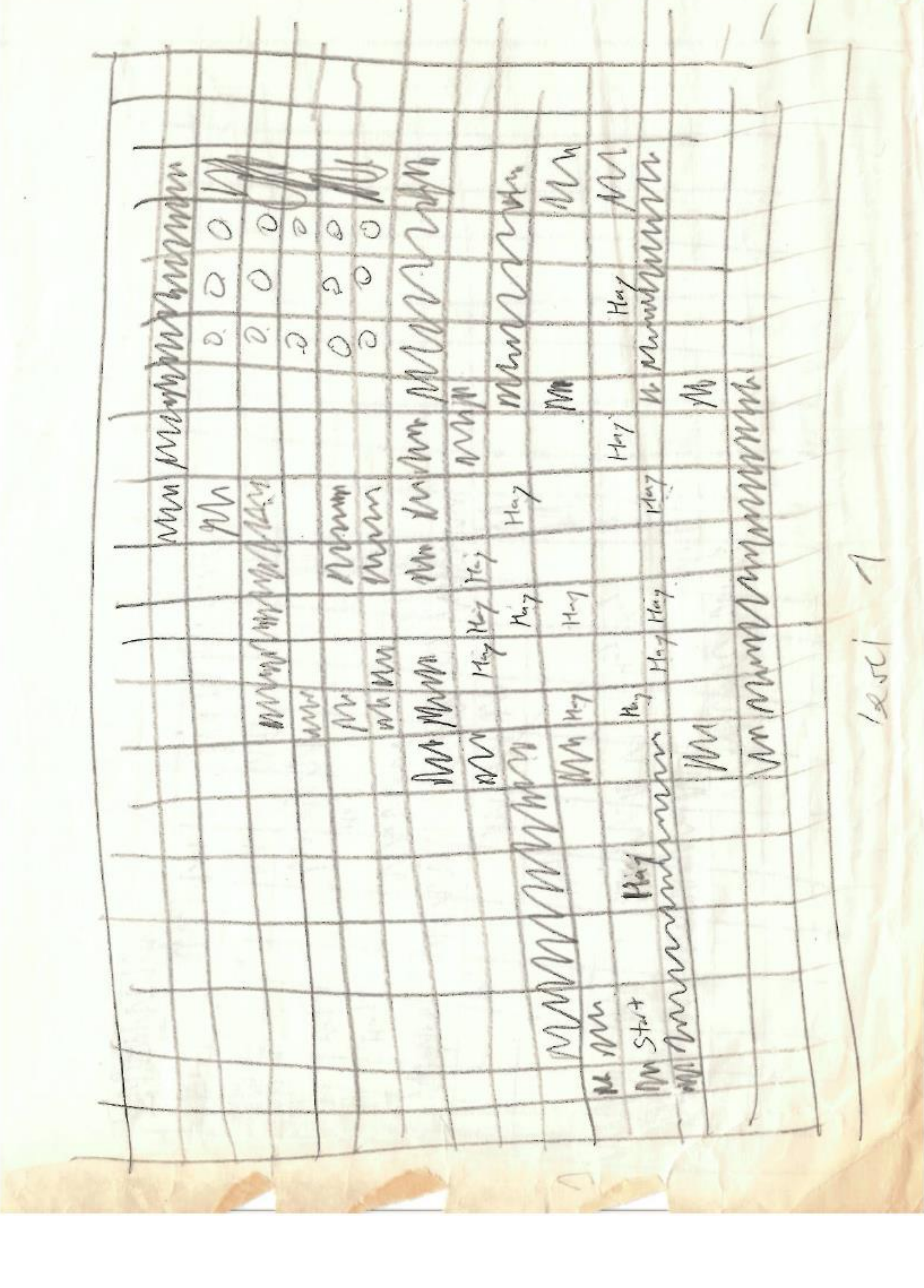


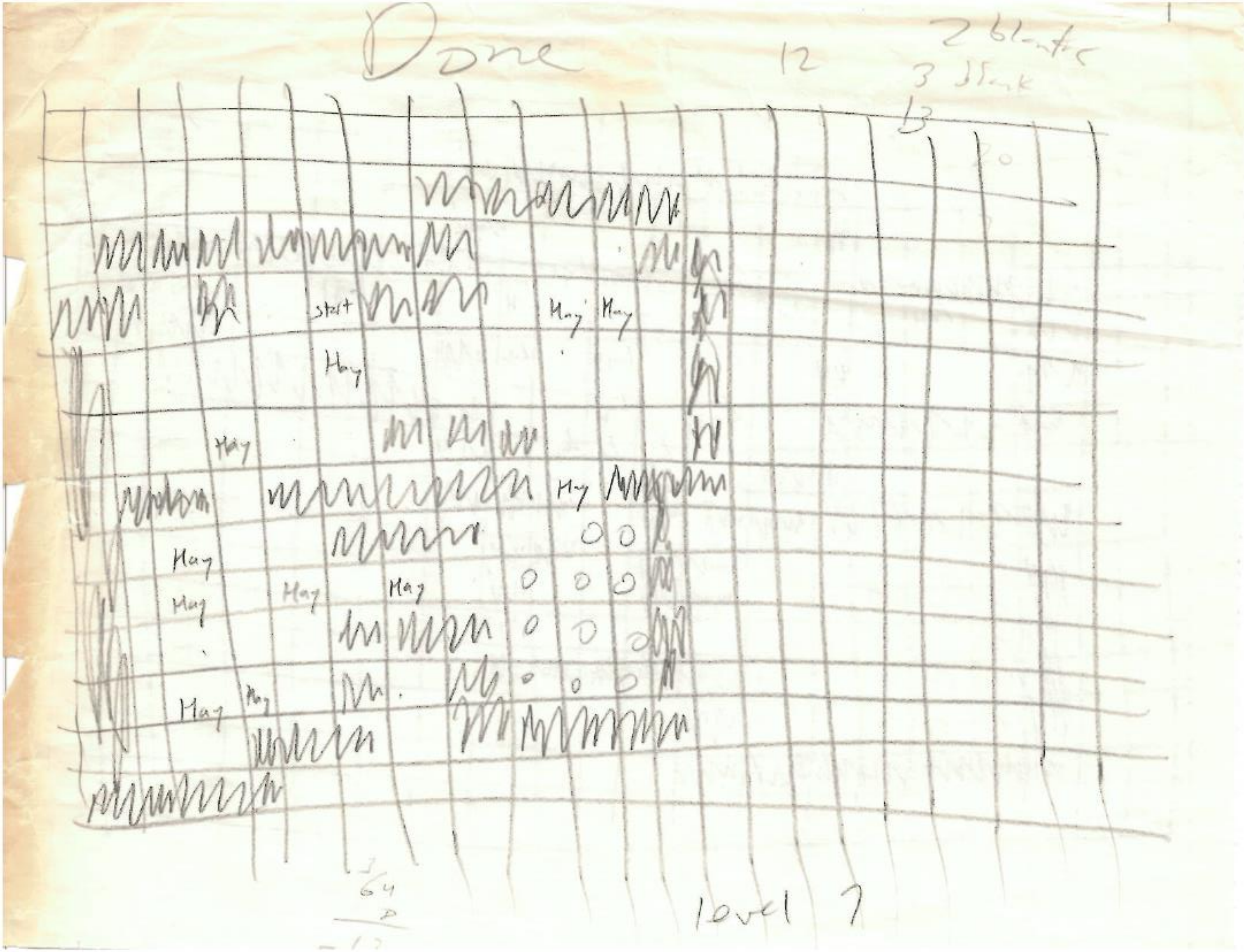
3
Start
Hay
Hay
Hay
Hay
Hay
Hay
Hay
Hay
Hay
Hay
Hay
Corral 3
Spice

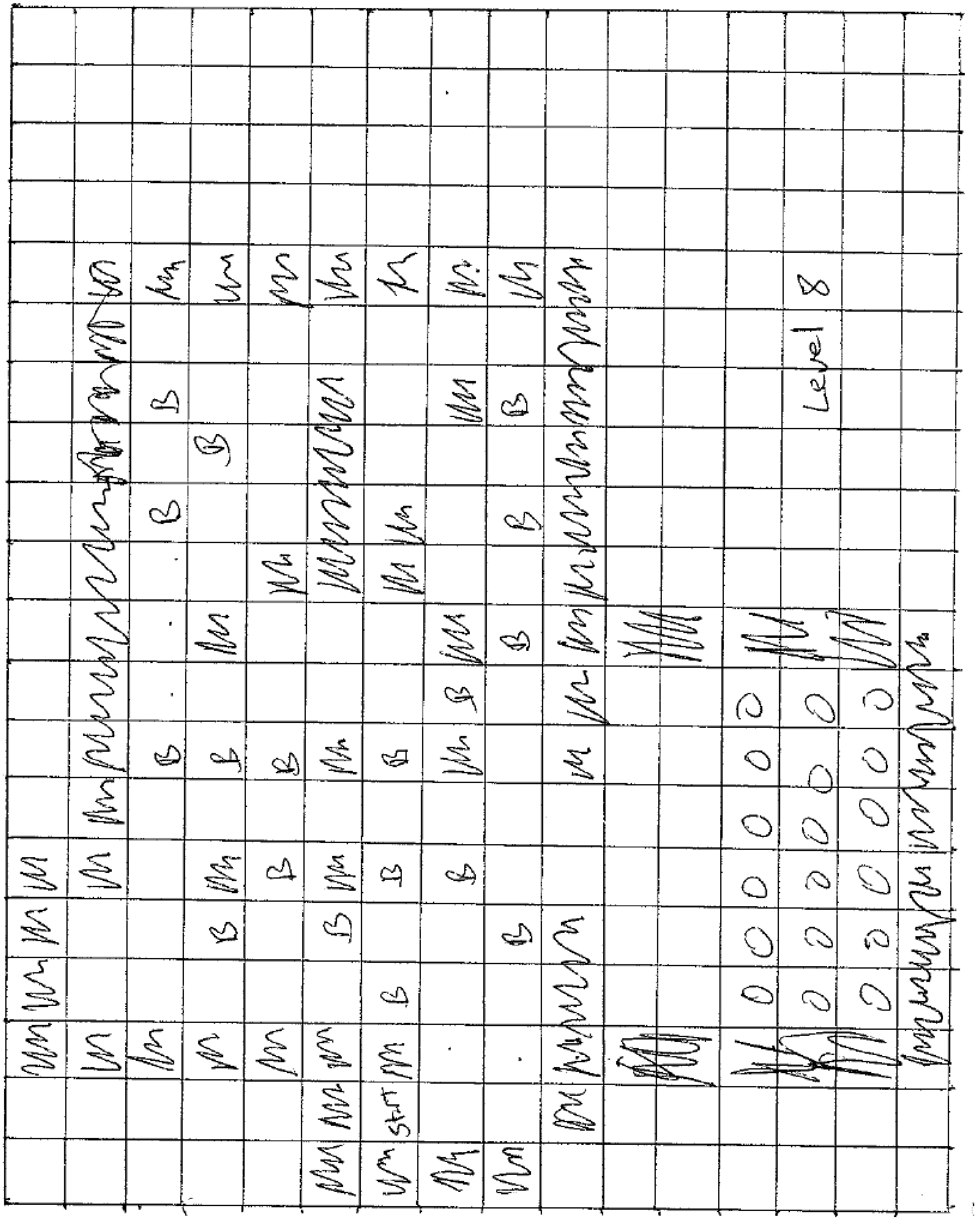

Level 4
Start

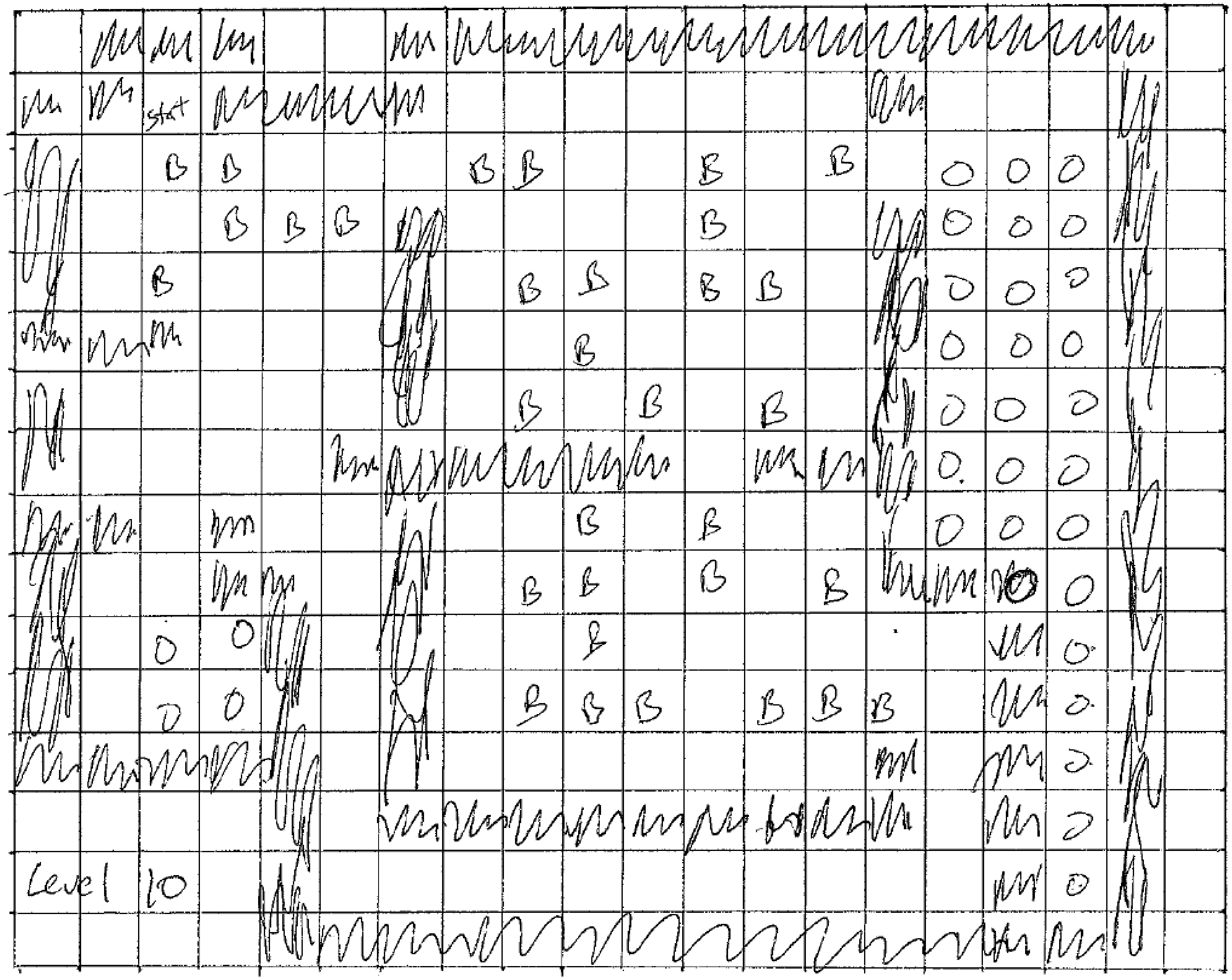

Level 5
B
O
Start
end

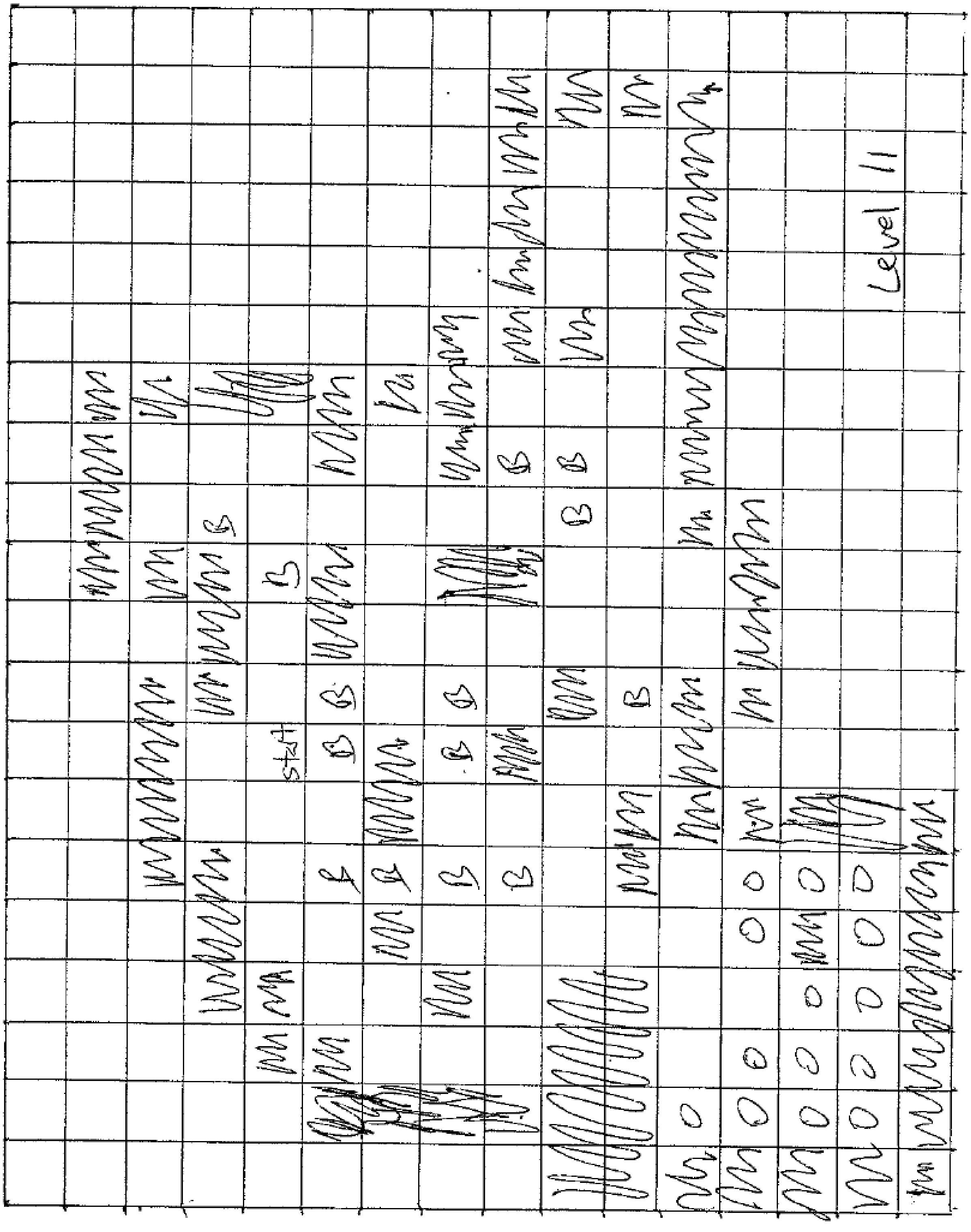

Level 6
start

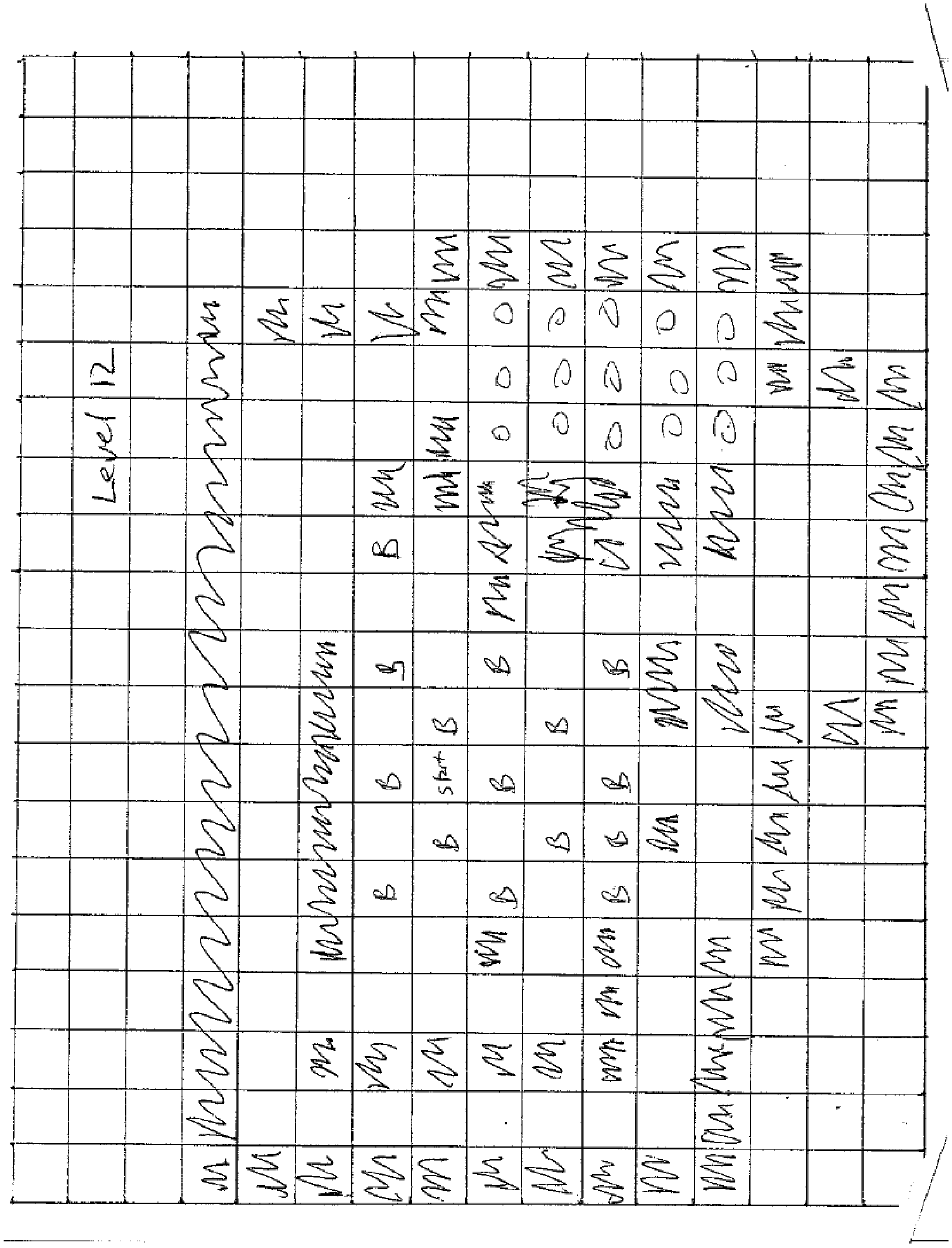

Start
Hay
Level 1

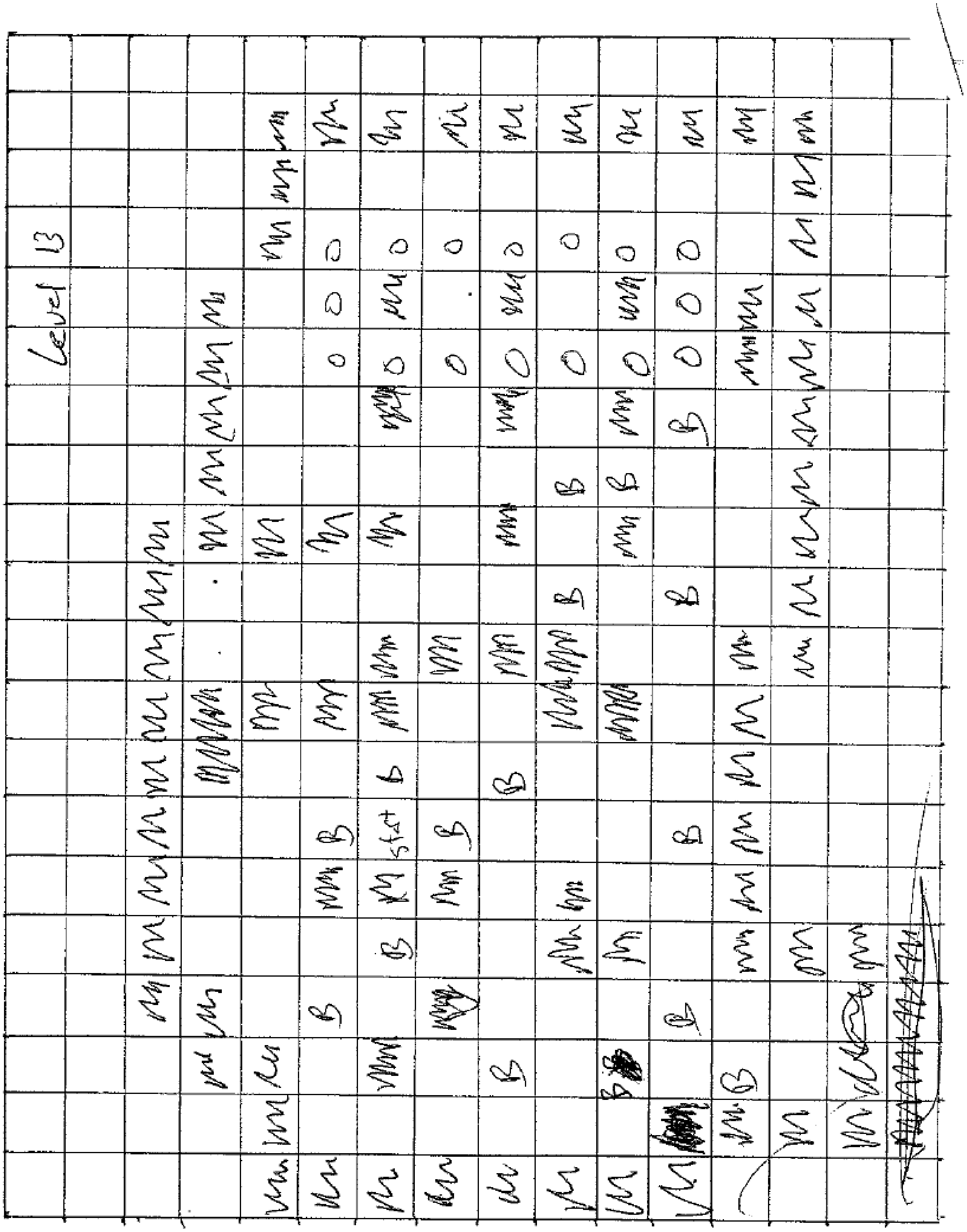

Done
12
2 blanks
3 Stack
13
20
start
May
May
May
May
May
May
May
May
May
May
May
May
3/64
8
12
level 7

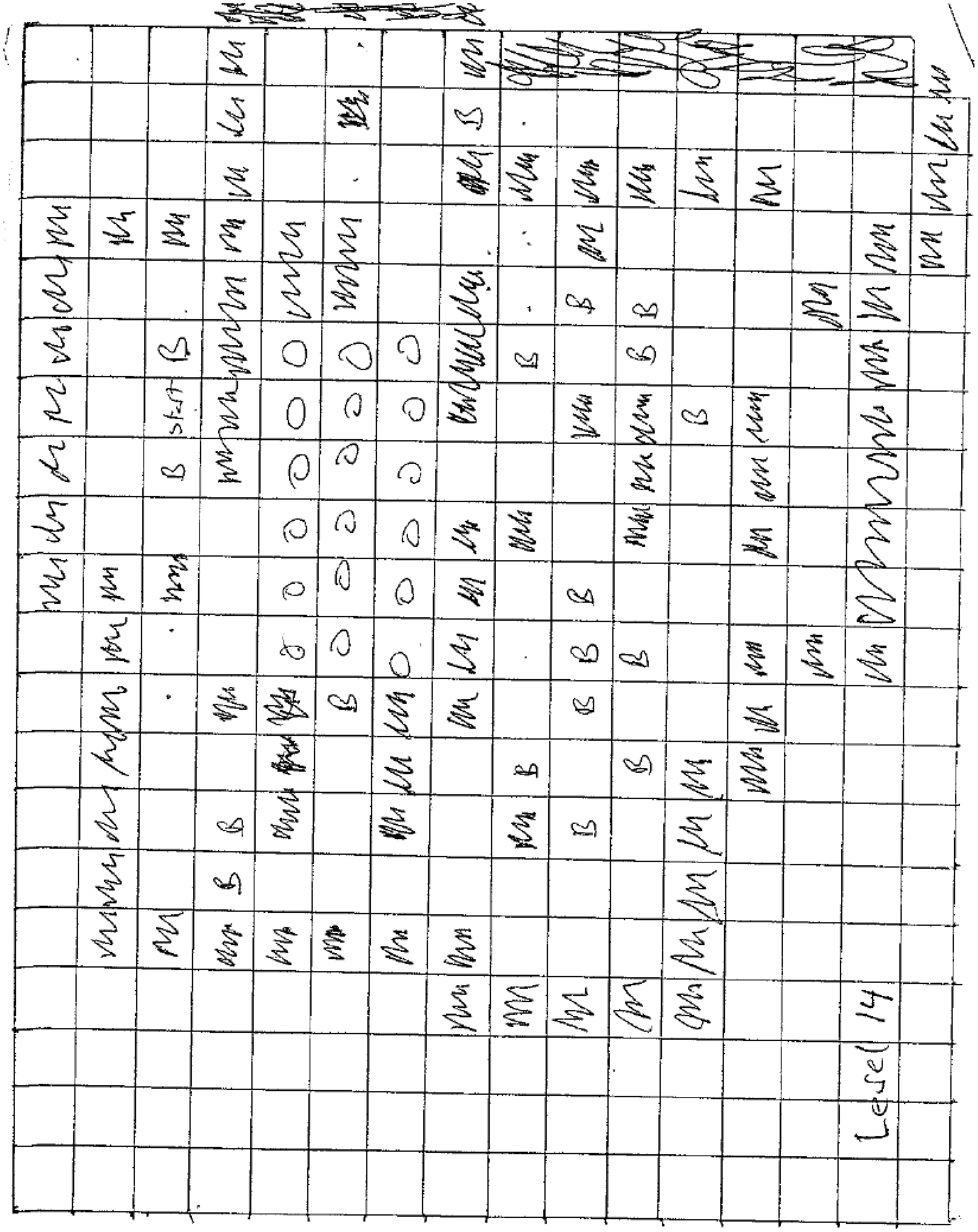

Level 8
start

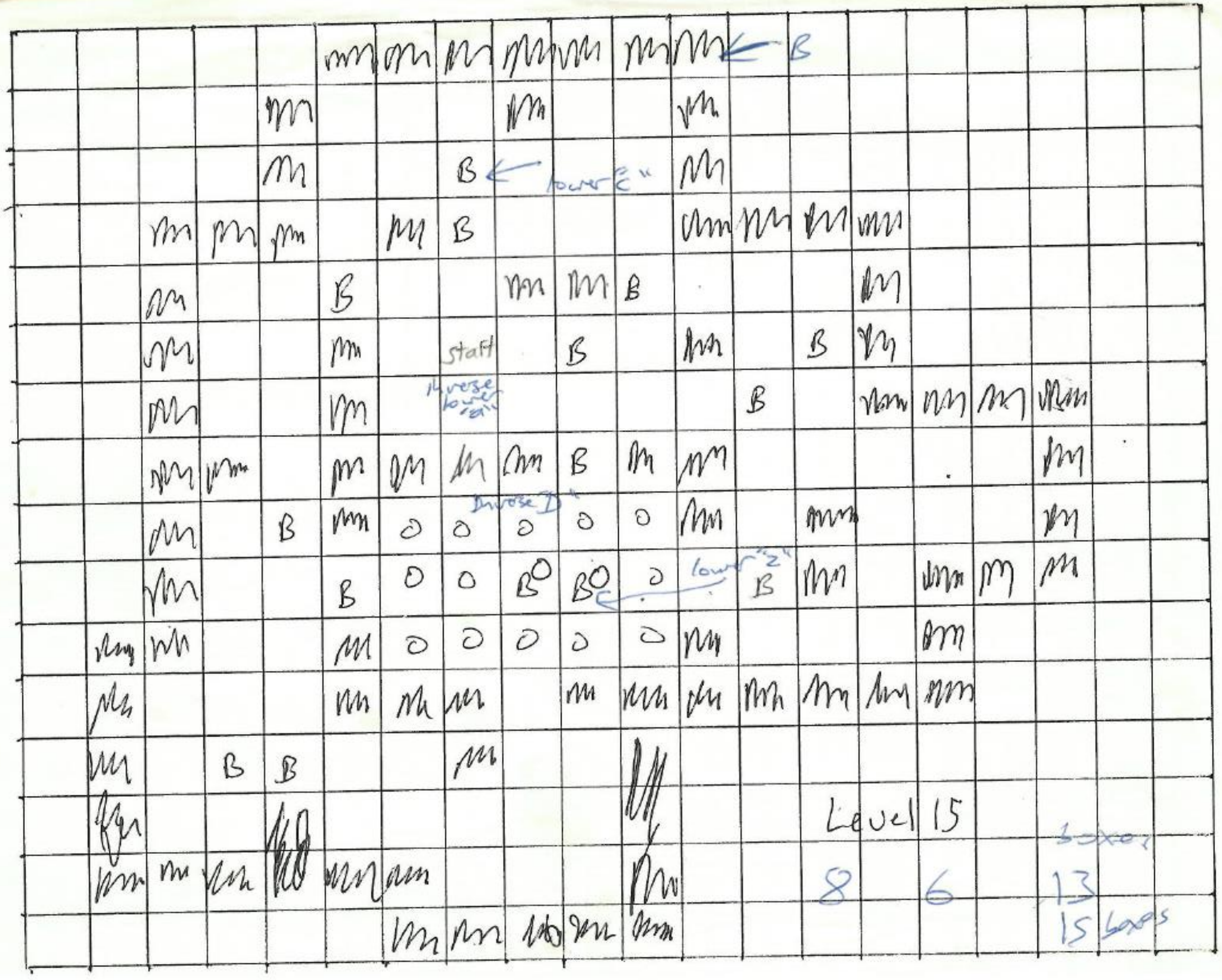

start
B
O
Level 10


Level 11
start
B
O


Level 12
start
B
O


Level 13
start
B
O


start
Level 14


B
lower "c"
staff
Inverse lower "a"
Inverse "D"
lower "Z"
B
O
Level 15
8
6
boxes
13
15 boxes

														Level 16	
						X	X	X	X	X	X	X			
					X	X	O	O	O	O	O	X			
		X	X	X	X			X	X	O	O	X			
		X		B		B	X		X	O	O	X			
		X					X		O	B	O	X			
		X	X			B		B	O	O	O	X			
			X				X	X	B	X	X	X			
			X	B	B		X	X				X			
	X	X	X		X	X		B			B	X	X	X	
X	X			B		X		X			B			X	
X				B	St		B		X					X	
X			B		X				X	X	X	X	X	X	
X					X	X	X	X	X						
X	X	X	X	X	X										

					X	X	X	X	X	X	X			Level 17	
				X	X			X			X	X			
				X								X			
	X	X	X	X			X	X	X	B		X			
	X			X	X			X				X			
	X							X	X	X	X	X			
	X						X	X	X			X			
	X	X		X	X		X		X			X	X	X	
	X				X				X	B				X	
	X				X			X	X		B	X		X	
	X	O	O	O	X		X	X				X		X	
	X	O	O	O	X					B				X	
	X	X	X	X	X			X	B		B	X	X	X	
					X	X	X	X		X	start	X			
								X				X			
								X	X	X	X	X			

					X	X	X	X	X	X	X	X	X		
					X								X		
				X	X	B	X	X	X	X			X		Level 18
	X	X	X	X			X			X		X	X		
	X			X					B	B		X			
	X		Start	X		X	X		B			X			
	X						X	X		X		X			
	X	X		X		B			B			X			
	X			O	X			B	X			X			
	X			O	X		X			X		X			
	X	O	O	X	X		X	X				X			
	X		X	X	O	O		B		X		X			
	X				O	O	O	X				X			
	X	X	X	X		O	O	X	X	B		X	X		
				X	X	X		B					X		
				X				X	X				X		
				X		B		X	X	X	X	X	X		
				X	X			X							
					X	X	X	X							

			Level 19							X	X	X	X	X	X
										X	O	O	O	O	X
										X	O	X	O	O	X
										X	O	O	O	O	X
										X	O	O	O	O	X
			X	X	X	X	X	X	X	X	X				X
			X											X	X
			X		X	X	X	X	X	X	X			X	X
	X	X	X		X							B	B		X
	X				X		X	X		X	X				X
	X			X	X		X				X		X	X	X
X	X			X	X		X		X		X		B		X
X	Start	B				B		B	B		X				X
X			X				B		X	X		B			X
X		X	X	X	X					X			B		X
X		X				B	B	X	X	X	B		X		X
X	X	X		B	B								X		X
		X				X	X	X	X	X			X		X
		X	X	X	X	X	X	X	X	X	X	X	X	X	X

Level 20															
	.	.	X	X	X	X	X	X	X	X	X	X	X	X	
			X	O	O	O	O	O	O	O	O	O	O	X	
	X	X	X	O	[illegible]	O	[illegible]	O	[illegible]	O	[illegible]	O	O	X	
	X												O	X	
	X	start		B		B		B		BO	[illegible]	BO	O	X	
X	X	X	X	X	X	X		X	X	X	X	X	X	X	
X			X						X				X		
X			X						X		B		X	X	
X		B	X	X	X		X	X	X	B				X	X
X			B				X		B		B		B		X
X			X	X								X	B		X
X			X	X	B	X	X	X	X	B	X	X			X
X		X	X				X					X			X
X			X				X				B	B			X
X	X		X		B	B		X							X
	X		X					X		B		X	X	X	X
	X		X	X	X	X	X	X	X	X		X			
	X											X			
	X	X	X	X	X	X	X	X	X	X	X	X			

	21														
	Level						X	X	X	X	X	X	X		
							X		O	O	O	O	X		
							X		O		O	O	X		
		X	X	X	X	X	X	X	O	O	O	O	X		
		X			B			B	O	O	X	X	X	X	
		X			B		B			X	X			X	
X	X	X	X			B		B						X	
X								B		X	X	B		X	
X			X	X	B	X	X	B	X				X	X	
X			X	X		X	X		X		X	X	X		
X			X	Start	B							X			
X			X	X	B	X	X	X		X		X			
X					B			X				X			
X			X	X				X	X	X	X	X			
X	X	X	X	X	X	X	X	X							

L

22															
Level	X	X	X	X	X						X	X	X	X	X
	X				X	X	X	X	X	X	X				X
	X					B				B			X		X
	X		B		X			B					X		X
X	X	X			X	X	X	O	X	X	B		X		X
X			B	B	X	O	O	O	O	X		B	X		X
X					X	O	O	O	O	X			X		X
X	X	B	X	X	X	O	O	O		X					X
	X			Start	X	O	O	O	O	B	B	B	B	X	X
	X		B		X	O	O	O	O	X					X
	X	X		X	X	O	O	O	O	X	X	B	X		X
	X				X	O		O	O	X	X		X		X
	X		B		X	X	X	B	X	X	X		X		X
	X	X			X					B			X		X
X		B		B					B				X		X
X			B		B		X			B			X		X
X				B		X	X	X	X		B	B			X
X				X									X		X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

																Level		23	
				X	X	X	X	X	X	X									
				X			X			X	X	X	X						
X	X	X	X	X		B	X	B		X			X	X					
X	O	O		X			X			X				X					
X	O	O		X		B	X	B		X			B	X	X	X	X		
X	O			X						X	B			X			X		
X	O	O				B	X			X		B					X		
X	O	O	start	X			X	B		X	B			X			X		
X	O	O		X		B	X						B	X			X		
X	O	O		X			X	B	B	X	B			X			X	X	
X	O	O		X		B	X			X			B	X	B			X	
X	O	O		X			X			X				X				X	
X	X	O		X	X	X	X			X	X	X	X	X				X	
	X	X	X	X			X	X	X	X				X	X	X	X	X	

24															
Level			X	X	X	X	X	X	X	X	X	X	X		
	X	X	X			X	X	X	X	X			X		
	X			B									X		
	X				X	X	X	X	X	X			X		
X	X	X					X	X	X		B	X	X	X	X
X	O	O	X		B		X	X	X						X
X		B				X				X	B				X
X		B	B			X						B		X	X
X	O	O	X	B	X		X		X	X		B			X
X	O	O	X				X			X	B	B			X
X	O	O	X					X				X		X	X
X	O	O	X	B		X	X	X	X	B		B	B		X
X	O	O	X		X			X			X				X
X	O	O	X		X			B	B					X	X
X	O	O	X		X		X	X				B	X	X	X
X	O	O	X		X		B				B			start	X
X	O	O	X		[illegible]			B	X	X					X
X	O	O	X		X				X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X					

X	X	X	X														Level		25
X			X	X	X	X	X	X	X	X	X	X	X	X	X	X			
X			X				O	O	X	O	O	O	O	O	O	X			
X			X		X		X	X	X	X	X		O	O	O	X			
X	X	B	X					O	O	O	O	O	O	O	O	X			
X				X	X	B	X	X	X	X	X	X			X	X	X	X	
X		B		X						X	X	X	X	X	X	Start		X	
X	X	B		X		B				X	X	X	X	X	X			X	
X			B		X	B	B	B	X	X								X	
X							X					X	B	X	B	X	X	X	
X		X	X	X	X		X	B	B	B	B	B					X		
X		X					B						X				X		
X		X				X	X		X	X						X	X	X	
X		X	X	X	X	X	X	B	X	X	X	X	X	X		B		X	
X									X					X				X	
X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	

	26														
	Level								X	X	X	X	X		
					X	X	X	X	X			Start	X	X	
					X			O	X	B		B		X	
				X	X				X			B		X	
				X	O	O	O	O	X	B	X	X		X	
				X	O	O	O	O	X			X		X	
				X	O	O	O	O	X		B			X	
			X	X	X	X		X	X		B			X	
			X				B		B			X	X	X	
			X				B					X			
			X	X	X	X	X	X	X		X	X			
			X			B		B				X			
			X				B			B		X			
			X	X	X					X	X	X			
					X	X	X	X	X	X	X				

		X	X	X	X	X	X	X	X	X	X				
	X	X					X				X				
	X		B				B		B		X			Level 27	
	X		X		X	B	X		X	B	X				
	X						X		X		X				
	X	X	X	X	X	B		B			X	X			
	X						B			B		X	X		
	X		X				X	X	X		B		X		
	X		X	B	X	B	X				X	Start	X		
	X		B				B		X		X		X		
	X	X				X	X		X		B		X		
	X			X	X	X			X			X	X		
	X		O	O	O	X		B		B		X			
	X		O	O	O	X			X	B		X			
	X	O	O	O	O	X			X			X			
	X	O	O	O	O	X				B	B	X			
	X	O	O	O	O	X	X	X				X			
	X	X	X	O	O	X		X	X	X	X	X			
			X	X	X	X									

		X	X	X	X	X	X	X	X	X	X	X			
		X	O	O				O	O	O	O	X			Level 76
		X	X	O	O	O	O	O	O	O		X			
X	X	X	X	X	O	O	O	O	X	O		X	X	X	
X				X	X	O	O	X	X	B	X			X	
X			B		X	X	X	X						X	
X		B			X			X			B	B		X	
X	X		B	B				X	B	B			X	X	
	X			B	X				B			B	X		
	X	B			X			X		B	B		X		
	X		B		X	X	X	X	B		B		X		
	X	X			X								X		
		X	X		X			B		X	X	X	X		
			X		X	X		X	X	X					
			X				B		X						
			X			X		Start	X						
			X	X	X	X	X	X	X						

	Level 29			X	X	X	X	X	X	X	X	X	X		
				X			X	X	X				X		
				X		B				B			X		
				X			X	X	X	X	B	X	X		
				X	X		X			X			X		
			X	X			X	O	OB				X		
			X			X	X	O	O	X			X		
			X		Start		X	O	BO	X		X	X		
			X		X	B	X	O	O	X	B		X		
			X		B		X	O	O	X			X		
			X		X		X	BO	BO	X			X		
			X		B		X	O	O	X	B	X	X		
			X					O	BO	X			X		
		X	X	X			X			X			X		
	X	X					X	X	X	X			X		
	X			X	X	X	X	X	X	X	B	X	X		
	X		B							B			X		
	X			X	X				X				X		
	X	X	X	X	X	X	X	X	X	X	X	X	X		

										X	X	X	X		
	X	X	X	X	X	X	X	X	X	X			X	X	30
	X			X				B	O	O	O	O	O	X	Level
	X		B			B		X	O	O	B	O	O	X	
	X			X	B	B		X	O	O	O	O	O	X	
	X		B			B		X	O	O	O	O	X	X	
	X			X	X	X	X	X	X	X	X	X	X		
	X	X		X	X	X	X	X				X	X		
	X			B						B			X		
	X				X	X	X	X	B				X		
	X			X				X		X	X	X	X	X	
	X	X	X	X		B		X		X				X	
				X		B					B	B		X	
				X	X	Start	X			X				X	
			X	X	X		X			X		X	X	X	
			X				X	X	X	X		X	X		
			X		B	B					B		X		
			X				X	X	X				X		
			X	X	X	X	X	X	X	X	X	X	X		

				X	X	X	X								
			X	X			X								
			X				X							Level 31	
			X			B	X	X	X	X	X				
	X	X	X	X		B					X				
	X			B		B	X		B		X				
	X				B		X	X	X		X	X			
	X			B			B		X		O	X			
	X	X	B		B				X		O	X			
	X		start	X	X	X		X		O	O	X	X		
	X		B				B	B		O	O	O	X	X	X
	X	X			X	X		X	X	O	O	O	O	O	X
X	X	X	X	B	X				X	O	O	O	O	O	X
X			B		B			B	X	X	X	O	O	O	X
X						X					X	X	X	X	X
X	X	X		B			B	B			X				
		X	X	X	X				X	X	X				
					X	X	X	X	X						

															Level 32
									X	X	X	X			
				X	X	X	X	X	X			X	X		
				X		B		B				X	X		
	X	X	X	X			B		B		B		X	X	X
	X		B		B				B	B					X
	X			X			B				B		B		X
	X	X		X	X	X	X	X		X	X	X	X	X	X
		X				X	O	O	O		X				
		X			B		O	O	O	O	X				
		X		B		X	O	O	O	O	X				
		X	X		B	X	O	O	O	O	X				
			X	X	Start	X	X	X	X	X	X				
				X	X	X									

														Level 33	
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	X					X	X			X	Start				X
	X		B	B		X		B		X	X	X	B		X
	X	X		B		X	B			X		B			X
	X	X									B				X
	X			X	X	X	X	X	X	X		X	X	X	X
	X		B	X					B	O	O	O	O	X	X
	X				B	B	X			O	O	O	O	O	X
	X			X		B		X	X		X	O	O	O	X
	X			X		B			B		X	O	O	X	X
	X	X	X	X							X	O	X	X	X
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

	Level 34														
					X	X	X	X	X	X	X				
				X	X						X	X	X		
			X	X		B			X				X		
		X	X	Start		X		X	X	X	X		X	X	
	X	X			X	X		B		X	X	B		X	
	X		B	X			B		B					X	
	X					B		B			X			X	
	X			X	X	X			B	X	X	X	B	X	X
	X	O	X	O	O		X	B		X					X
	X	O	O	O	O		X			X		B	B		X
	X	O	X	O	O		X		X				X	X	X
	X	O	O	O	O			B					X		
	X	X	X	X			X	X	X	X	X	X	X		
				X	X	X	X								

X	X	X	X	X	X	X	X	X	X					Level 35	
X	O	O	O	O	X				X	X	X				
X	O	O	O	O	X		X				X				
X	O	O	O	O	X		X				X				
X	O	O	O	O							X				
X	X	O			X		X	X	X	X	X	X	X	X	X
	X		X												X
X	X			X	X	X	X	X	X	X	X				X
X		Start					X	X	X			B	B		X
X	X						X				X	B			X
X			X	X		X	X		X		X		X	X	X
X			X	X		X							X		
X		X	X	X		B		B		B	X		X	X	X
X		B			B		X	X	B		X				X
X	X	B	X	X		B				B			X		X
X				X	B		X	X			B	B	X		X
X							X		B	X		B			X
X				X	X	X	X						X	X	X
X	X	X	X	X			X	X	X	X	X	X	X		

					X	X	X	X						Level 36	
			X	X	X			X	X						
X	X	X	X			B			X						
X				B		B			X	X	X				
X		B				X		B				X	X	X	X
X			X			X				B		X	O	O	X
X	X	B	X	B		X	X	X	X	B	X	X	O	O	X
	X				X	X	X	X	X		X	O	O	O	X
	X	B	X		X	X	Start	X	X				O	O	X
	X		X					B				O	O	O	X
	X				X	X	X	X		X		O	O	O	X
	X	X	X		X	X	X	X			X	X	O	O	X
		X	X	B		X	X	X	X	B		X	O	O	X
		X				X	X					X	O	O	X
	X	X		B	B	X	X			B		X	X	X	X
	X						B	B	B	B		X			
	X		B		X	X	X					X			
	X				X		X	X	X	X	X	X			
	X	X	X	X	X										

	X	X	X	X	X	X	X	X	X	X					
	X									X					
	X					B		B		X				Level 3	
	X				X	X	X	X	X	X					
	X	X		X			X	X	X	X	X	X	X		
	X	X		X			B			X			X		
	X		B	X		B		B					X		
	X					X		X		X	B		X		
X	X	X			X		B	B		X		X	X	X	
X				X	X			X	X				SPLIT	X	
X					X		B			B	B	X	X	X	
X			B	B	X	X			X				X		
X	O	O			X		B		X	B			X		
X	O	O	X	B	B		B	B					X		
X	O	O	X		X				X			X	X		
X	O	O	X	O	O				X	X	X	X			
X	O	O	O	O	O	X	X	X	X						
X	O	O	O	O	O	X									
X	X	X	X	X	X	X									

														38	
														Level	
								X	X	X	X				
		X	X	X	X	X	X	X			X				
		X	start		B				B		X				
		X				X		X	X		X				
		X	X	B	X	O	O	O			X				
			X		X	O	O		X	B	X				
			X		B	O	O	O			X	X			
			X			X	B	X				X			
			X			B				B		X			
			X			X	X	X	X	X	X	X			
			X	X	X	X									

X	X	X	X	X	X	X	X	X	X						
X	O	O	O	O	O	O	X		X						Level 39
X	O	O	O	O	O	O	X		X	X	X	X			
X	O	O	O	O	O	O	O	O				X			
X	O	O	O	O		O	X		X			X			
X	X	X	X		X		X		X		X	X			
	X	X	X						X		X				
	X		Start	X	X	X	X	X	X		X	X	X	X	
	X			B			X	X	X		B			X	
	X		B		B		X						B	X	X
	X			B	B	B		B	X	X	B		B		X
	X			B			X					B			X
	X	X	X		X	X	X				B				X
	X	X	X		X	X	X	B	X		X	X	X	X	X
	X		B				X		X	B		B		X	
	X		B		B	B	X		X		B	B		X	
	X								B			B		X	
	X	X					X	X						X	
		X	X	X	X	X	X	X	X	X	X	X	X	X	

	X	X	X	X											
X	X		Start	X	X	X	X	X	X	X	X			40	
X											X				
X		B	X	X	X	X	X	B	X		X			Level	
X			X				X		B		X				
X		B		B					B	X	X				
X		B			B			X			X				
X		X	X	X	X	B		X	X		X				
X			B				B		X		X	X	X	X	X
X		X	X	X	B	X				B		X			X
X				X			B	X		B					X
X			B							X	X	O			X
X	X	X	X	X			X			X	O	O	O	O	X
				X	X	X	X	X	X	X	O	O	O	O	X
										X	O	X	O	O	X
										X	O	O	O	O	X
										X	X	X	X	X	X

						X	X	X	X	X	X	X	X	.	
						X		Start	X	X			X	X	
	X	X	X	X	X	X		B		B		B		X	
X	X					X		B						X	
X				B	B	X		B		B			X	X	
X			B				X		X	X	X	X	X		
X			B			B	X		X			X			
X	X			B		B				O	O	X			
	X	X		B			X		X	X	O	X			
		X	X		B		X		X	X	O	X			
			X	X			X		X	X	O	X			
				X	X	X	X		X	X	O	X		41	
							X		X	X	O	X			
							X		X	X	O	X		Level	
							X		X	X	O	X			
						X	X				O	X			
						X			X	X	O	X			
						X		O	O	O	O	X			
						X	X	X	X	X	X	X			

		X	X	X	X	X	X	X	X						
		X	Start	X	X				X	X	X	X			
		X		B				B				X		Level 4/2	
		X			B		B		B	B	B	X			
		X		B	B	X		X	[illegible]			X			
	X	X	B					B	[illegible]			X			
	X			B			B	B	B	B	B	X	X		
	X		B	X	X	X	X		X				X		
	X			B	O	O	O	O	X				X		
	X		X	X	O	O	O	O	X	B	B		X		
	X		X	X	O	O	O	O				X	X		
	X				O	O	O	O	X			X			
	X	X		X	O	O	O	O	X	B	B	X		2.	
		X		X	O	O	O	O	X			X			
		X							B			X			
		X	X	X	X		X	X		X	X	X			
					X					X					
					X	X	X	X	X	X					

							X	X	X	X	X	X	X		
				X	X	X	X			X			X	Level 43	
				X			X						X		
				X				B		X	B		X		
			X	X		X	X		X	X		X	X		
			X			X	X			X			X		
			X		B			B			B		X		
			X	X		X			X	X			X		
			X			X	X		X			X	X		
			X					B		Start	X	X			
			X	X	X	B	X	X	X		X				
				X		B		B			X	X			
				X		X						X			
				X				X				X			
				X	X	X		X		X	X	X			
						X		X	O	O	D	X			
						X			O	O	D	X			
						X	X		O	O	D	X			
							X	X	X	X	X	X			

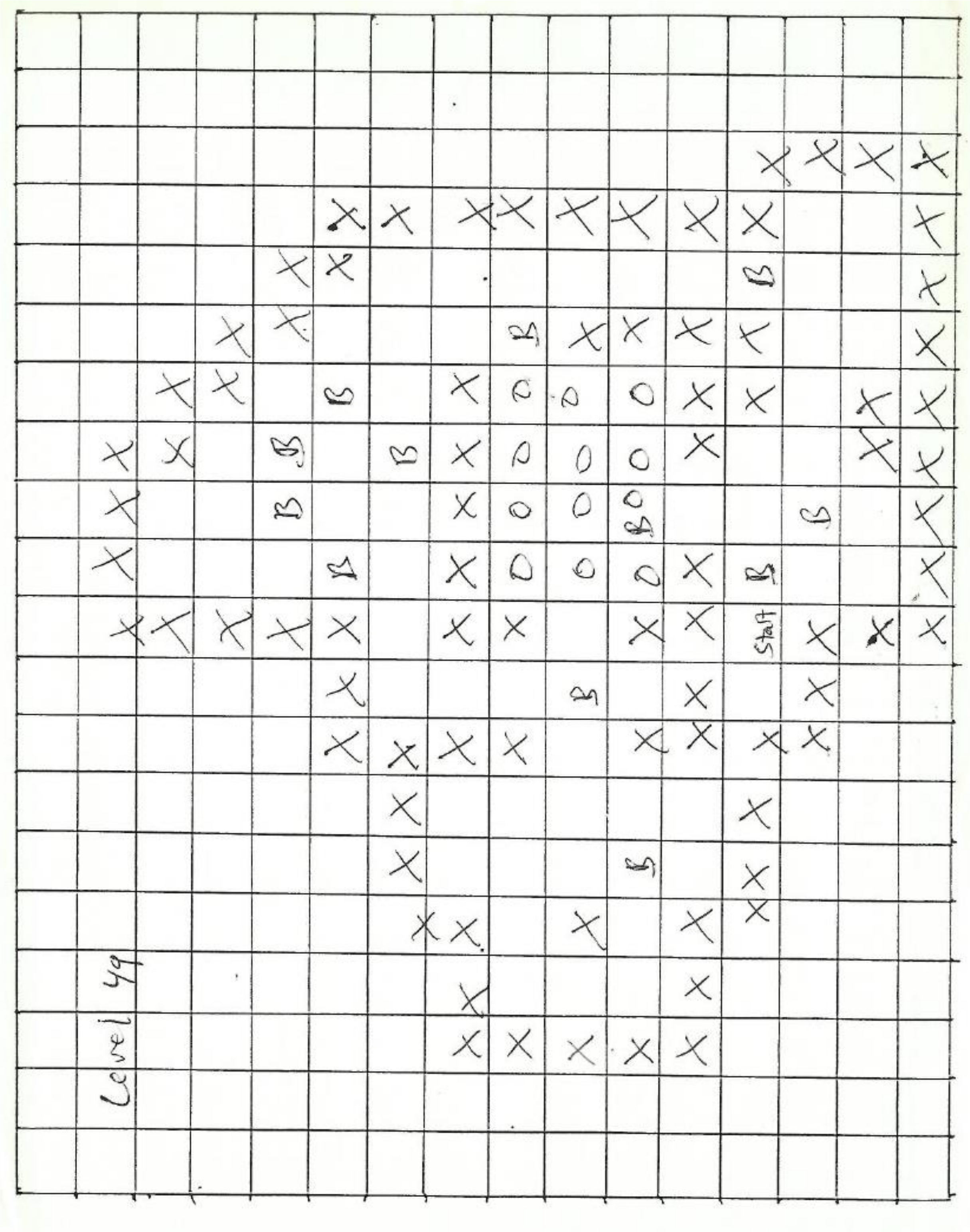
															Level		44		
	X	X	X	X	X														
	X				X														
	X		X		X	X	X	X	X	X	X								
	X							B	Start	X	X	X	X	X	X				
	X		B		X	X	B		X	X	X				X				
	X		X	X	X	X		B					B		X				
	X		X	X	X	X	X		X			X	B		X	X	X	X	
X	X			X	X	X	X		X	X	B							X	
X			B	X			B			X		X	X		X	X		X	
X										X		X	O	O	O	X		X	
X	X	X	X	X	X			X	X	X			O	O	O			X	
					X	X	X	X		X		X	O	O	O	X		X	
										X		X	X	X		X		X	
										X								X	
										X	X	X	X	X	X	X	X	X	

								X	X	X	X			Level 45	
				X	X	X	X	X			X				
				X				X			X	X			
		X	X	X		B		X		B		X			
		X		B	B		B				B	X	X		
		X				B		X					X		
	X	X	B					X	B		B		X		
	X			B		X	X	X			B	X	X		
	X				X	X			B	B		X			
	X	X	X	X	X			X	X		B	X			
			X	X	O	O				Start		X			
	X	X	X	O	O	O		X	X		X	X			
	X	O	O	O	O	O	X	X		B		X	X		
	X	O	O	O	O	X	X		B				X	X	
	X	O	O	9	X	X								X	
	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

		X	X	X	X										
	X	X			X	X	X	X	X	X	X	X	X		
	X		B		X	X	Start	X					X		
	X					B		X		X	X		X	X	X
X	X	X		B			B		B		X		O	O	X
X			B	X	X	X		B			X		O	O	X
X		X			B						X		O	O	X
X		X	B		B	X		B		X	X		O	O	X
X		X				X	X		X	X			O	O	X
X		X	B				B						O	O	X
X		X			B		X	X	X	X	X		O	O	X
X		X	X		X	X	X				X	X	X	X	X
X					X										
X	X	X	X	X	X										
														Level 46	

				X	X	X	X	X	X	X					
				X	O	O	O	O	O	X					
				X	O	O	O	O	O	X					
				X	O	O	O	O	O	X					
			X	X	X	X			O	X				Level 47	
		X	X		X	X			X	X					
		X		B		B			X	X	X				
		X				X	X	B			X				
		X	X			B		B	B		X				
			X		X	B		X			X				
			X		X		Start	X			X				
			X			B		X	B		X				
			X		X				B		X	X			
			X	B	X	B		X				X			
			X			B		B		B		X			
			X			X	X		B	X	X	X			
			X	X						X					
				X	X	X	X	X	X	X					

		Level 48													
								X	X	X	X				
								X		B	X	X	X	X	X
X	X	X	X	X	X	X		X		B		B			X
X	O B	O	O B	O	O B	X		X			B				X
X	O	B O	O	O B	O	X	X	X		B		B		X	X
X	O B	O	O B	O	O B			X			B		B	X	
X	X	B O	O	O B	O					B		B	B	X	
X	O B	O	O B	O	O B			X			B		B	X	
X	O	B O	O	O B	O	X	X	X		B		B		X	X
X	B O	O	B O	O	B O	X		X			B				X
X	X	X	X	X	X	X		X		B		B			X
								X			X	X	X	X	X
								X	X	X	X				



												X	X	X	X
					X	X	X	X	X	X	X	X			X
				X	X							B			X
			X	X				B	X	X	X	X			X
		X	X		B		X	O	O	O	X	X		X	X
	X	X		B		B	X	O	O	O	X			X	X
	X			B			X	O	O	O B			B		X
	X				B		X	O	O	O	X	B			X
	X	X	X	X	X		X	X		X	X	Start	X	X	X
					X				B		X		X		
					X	X	X	X		X	X	X	X		
						X						X			
						X				B		X			
						X	X		X		X	X			
	Level 49						X				X				
							X	X	X	X	X				

										X	X	X	X	X	
X	X	X	X	X	X	X	X			X				X	X
X			X				X	X	X	X					X
X		B			X					X		X			X
X			B	B			X		X	X					X
X			X												X
X				X			X	X	X	X			X	X	X
X			X								X	X	X		
X		B		X	X			X	X	X	X				50
X			X				X				X				
X	O	BO	O	X			X		X		X				Level
X	O	O	O B	O	X		X				X	X	X		
X	X	O	O	O B	O	X		B		B			X		
	X	X	O	O	O		B				X		X	X	X
		X	X	O	O	X			B		B				X
			X	X	X	X	B		B			B			X
					X		Start	B	X	X	X	X	X	X	X
					X				X						
					X	X	X	X	X						

X	X	X	X	X	X	X									
X						X	X		X	X	X	X	X		
X		X	X	♡			X	X	X	O	O	O	X	X	15 [illegible]
X				♡	♡			X	O	O		O	O	X	
X	X	X		♡				X	O	O		O	O	X	
		X	♡		X	♡			O	O	●	O	O	X	
		X					♡	X	X	O	O	O	O	X	
		X		X	♡	♡			♡		X	X	X	X	
	X	X					X		X	O	X				
X	X		♡		X	X	X		♡		X	X	X		
X	S			X	X				X				X		
X			X	X	X		X		X	X	X		X		
X		♡							♡		X		X		
X	X	X	X	X	X	X	♡		X		X		X	X	
X	X				X			X	X		X			X	
X		♡			♡		♡				♡			X	
X			♡		X			X	X	X		X	X	X	
X			X	X	X	X						X			
X	X	X	X			X	X	X	X	X	X	X			

		X	X	X	X	X	X	X	X	X	X	X			
	X	X										X		Level 2	
	X		O	O	O			X		X		X			
	X		O	O	X			X		O	O	X			
	X	X	O	X	X		X	X	X	O	O	X			
	X			B		X		B		O	O	X			
	X		B			B			B		X	X			
	X		B		B		B		B			X			
	X	X		B				B	B		X	X			
	X		B		B		B		B			X			
	X		B			B			B			X	X	X	
	X		X	B	B	B	B	B			O	O	O	X	
	X						X			X	O	O	O	X	
	X	X		X		X	X	X			O	O	O	X	
		X		S		X		X	X		O	O	O	X	
		X	X	X	X	X			X	X	X	X	X	X	

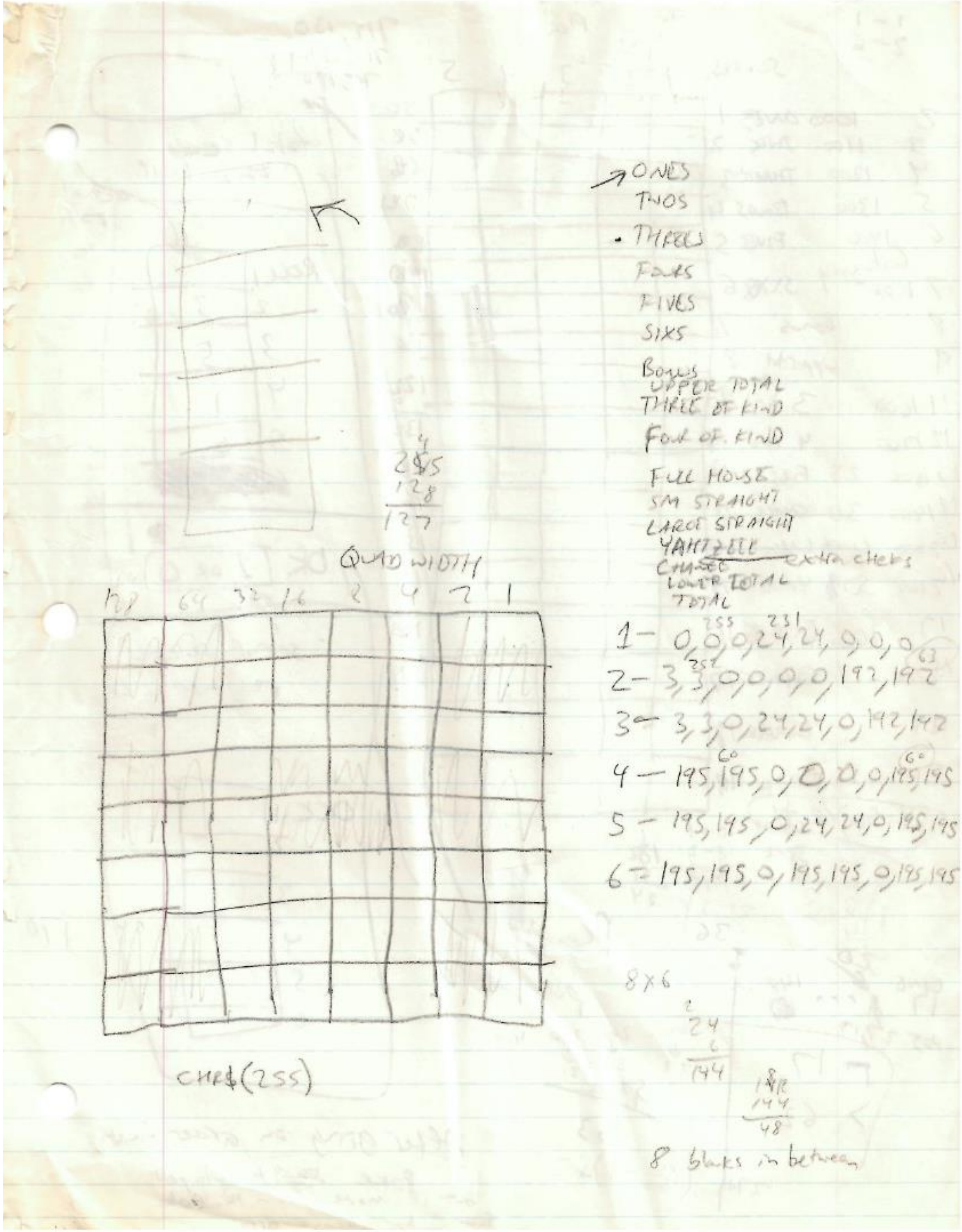
							X	X	X	X	X	X	X		
							X						X		
							X		X				X		LEVEL 5
							X						X		
							X				X	X	X		
X	X	X	X	X	X	X	X	X					X		
X	S		X	X	X			X		B			X		
X		B	B					X	X	X	B		X		
X				X			X	X		B			X	X	X
X	X	B		B				O	O	X			B		X
	X		X		X		X	O	O	O	B	X	X		X
	X		B			B	X	X	O	O	O	O	X		X
	X			X	B		X		X	O	O	O	X		X
X	X		B			X	X			X	O	O	X		X
X			X				X				B	O	X		X
X				B			X				X				X
X			X	X	X	X	X				X	X	X	X	X
X	X	X	X				X				X				
							X	X	X	X	X				

		X	X	X	X	X									
		X				X	X	X	X	X	X	X	X		
		X		B							X	S	X		
		X	B	B		X		B	B	B			X		
		X		X	X	X		X			B		X		
		X		X	O	O	O	X	X				X		
		X		X	O	O	O	X	X	X	X	X	X		
		X		X	O	O	O	X					X		
		X		X	O	O	O	X			B		X		
		X		X	O	O	O	X		B	B	X	X		
	X	X		X	X		X	X	X			X			
	X						X	X	X	X		X			
	X		X		B		X		B			X			
	X			X		X	X			B		X	X		
	X		B				B						X		
	X	X	X				X	X	X	X			X		
			X	X	X	X	X			X			X		
										X	X	X	X		

LEVEL 54

					X	X	X	X	X	X	X	X			
					X							X			
					X	B		B		B		X		LWB 55	
					X		X	X	B			X			
					X		X		B			X			
					X		X		B			X			
					X		X	S	B	X		X			
					X		X		B			X			
					X		X		B			X			
				X	X		X	X	X		X	X			
				X				X			X				
				X	O	O	O			X	X				
				X	O	O	O	X	X	X					
				X	O	O	O	X							
				X	X	X	X	X							

YAHTZEE



ONES
TWOS
THREES
FOURS
FIVES
SIXS
BONUS
UPPER TOTAL
THREE OF KIND
FOUR OF. KIND
FULL HOUSE
SM STRAIGHT
LARGE STRAIGHT
YAHTZEE ———— extra checks
CHANCE
LOWER TOTAL
TOTAL

255
128
127

QUAD WIDTH

128 64 32 16 8 4 2 1

1 — 0, 0, 0, 24, 24, 0, 0, 0
2 — 3, 3, 0, 0, 0, 0, 192, 192
3 — 3, 3, 0, 24, 24, 0, 192, 192
4 — 195, 195, 0, 0, 0, 0, 195, 195
5 — 195, 195, 0, 24, 24, 0, 195, 195
6 — 195, 195, 0, 195, 195, 0, 195, 195

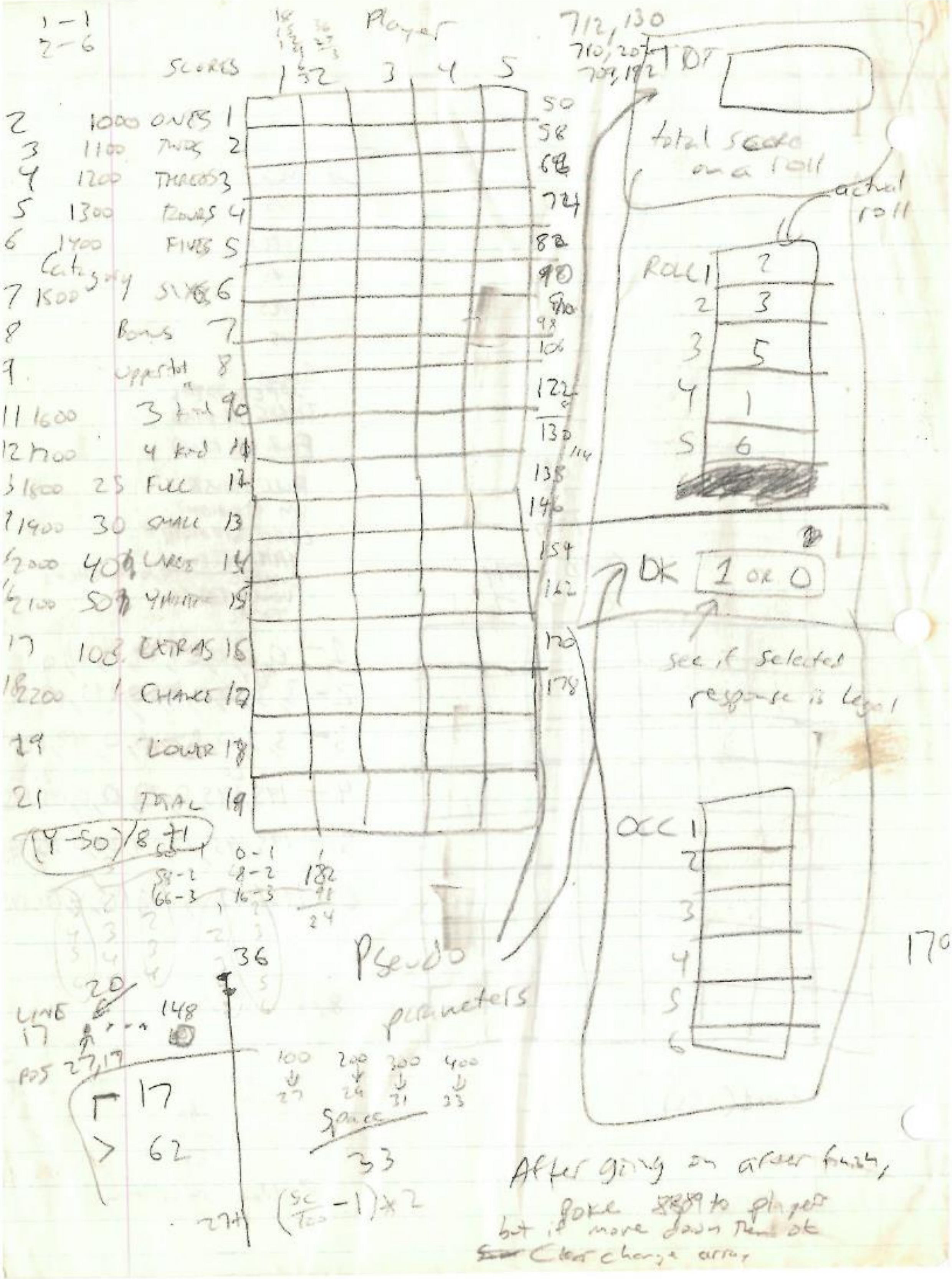
CHR$(255)

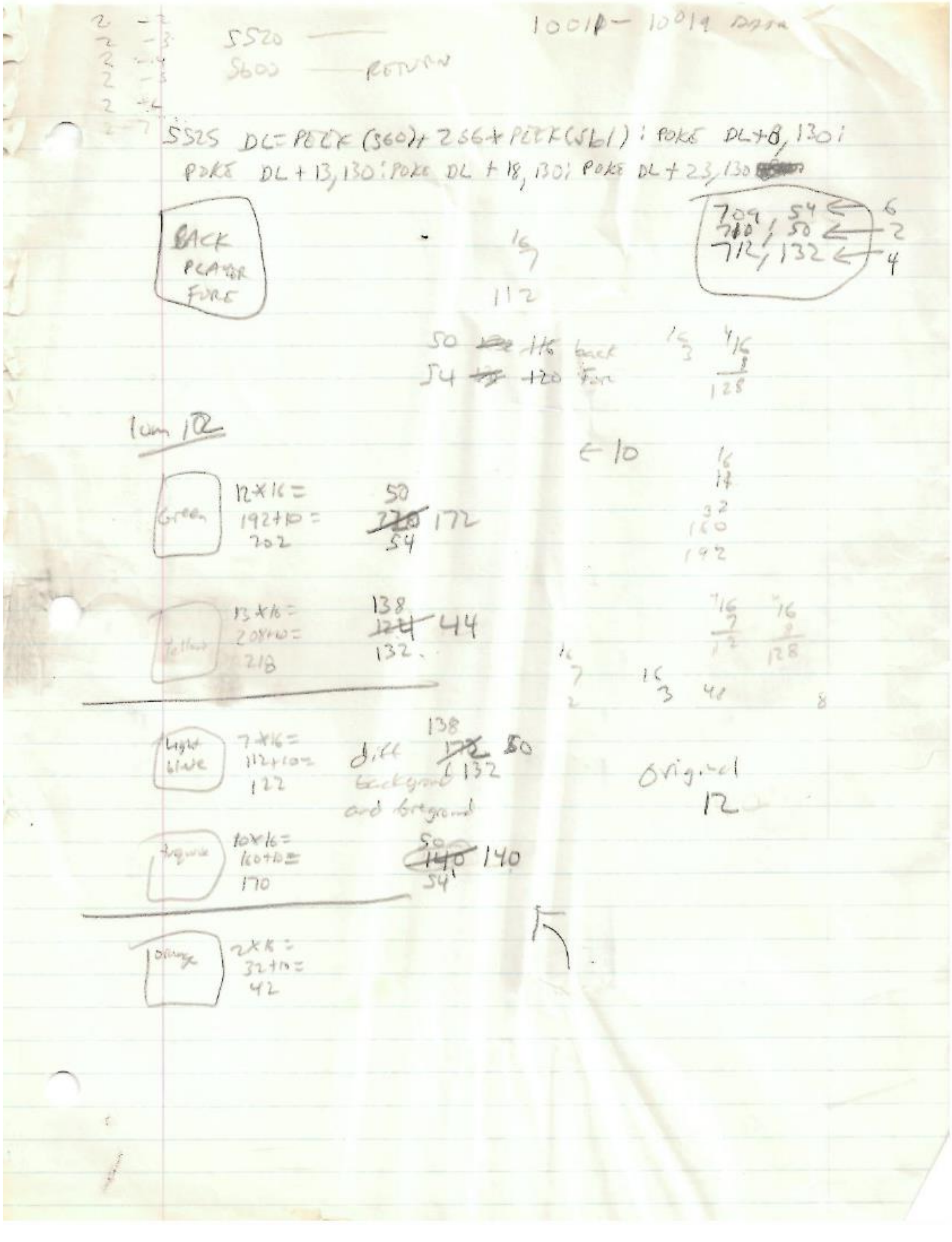
8 x 6

24
6
144

148
144
48

8 blanks in between





2 -2
2 -3
2 -4
2 -3
2 -4
2 -7

5520 ——

5600 —— RETURN

~~10018~~ — 10019 DATA

```
5525 DL=PEEK(560)+256*PEEK(561):POKE DL+8,130:
  POKE DL+13,130:POKE DL+18,130:POKE DL+23,130
```

BACK
PLAYER
FORE

709, 54 ← 6
710, 50 ← 2
712, 132 ← 4

16
7
112

50 ~~[illegible]~~ 116 back
54 ~~[illegible]~~ 120 Fore

16
3

16
8
128

Lum 12

← 10

16
12
32
160
192

Green
12*16=
192+10=
202

50
~~220~~ 172
54

Yellow
13*16=
208+10=
218

138
~~124~~ 44
132.

16
7
112

16
8
128

16
7
2

16
3 48

8

Light blue
7*16=
112+10=
122

diff background and foreground

138
~~172~~ 50
132

Original
12

Turquoise
10*16=
160+10=
170

50
~~140~~ 140
54

Orange
2*16=
32+10=
42

20
●
⊟ 18

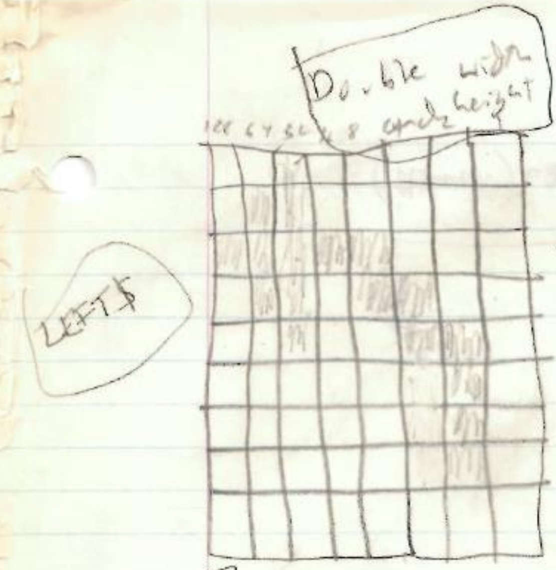
~~85, 85, 85, 85, 85, 85, 85~~

21, 21, 21, 21, 21, 21, 21
~~18, 18, 18, 18, 18, 18, 18~~
32, 32, 124, 32, 124, 32, 32
32, 6, 13, 13, 13, 7, 32
32, 22, 111, 32, 111, 2, 32
64, 22, 32, 62, 32, 2, 64
smile 32, 22, ~~7~~ 32, ~~14~~ 18, ~~6~~ 32, 2, 32
32, 7, 14, 14, 14, 6, 32
32, 6, 32, 32, 32, 7, 32
6, 32, 32, 123, 32, 32, 7

RED $ = [illegible]
SMILE $ =
WALK $ =

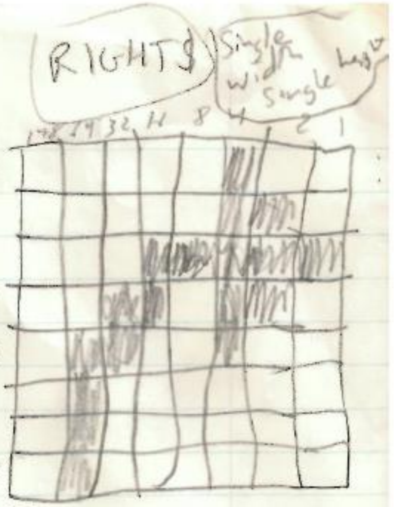
Right now, it stops after 13 turns, but this is wrong if you got a Yahtzee bonus. One person can end and the other can have games in hand.

~~When switching to new player, points don't show up in bonus Yahtzee section. well, they do, but then they are taken off.~~
~~Somehow I got into an infinite loop after printing second player's totals at end.~~



248
128

CHR$		CHR$	
32	64	36	4
16	96	38	6
248	248	63	31
108	108	86	54
38	70	100	100
2	34	0	64
2	34	0	64
2	34	0	64



```
DIM BL8$(8): BL8$(1)= CHR$(32): BL8$(8)= CHR$(32):
BL8$(2)=BL8$.

DIM BL16$(16): BL16$(1,8)= BL8$: BL16$(9,16)=
BL8$
```

34 34
1 3
32

POS

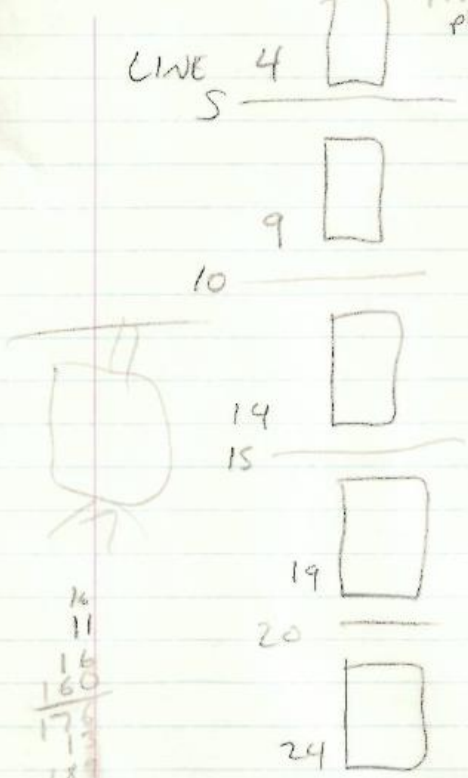
32

(35 - LEN(NAME$))

BLANK$(8)

2
35
16
19

~~709, 192 text~~
710, 202 back COLORS
712, 130 border
player 1



UPPER TOTAL

3 or 4 lines

14
16
84
140
224

232

LOWER TOTAL

BACK
PLAYER
FARE

16
160
174
14
188
214
6
208
256
6
1599
1536
63

192
L10

change player 1
change background
change text

6
11
16
66
110
176

14
16
84
140
24

16
11
16
160
176
13
189

change background 1

710,202

1600
X-COUNT
0
=1544

1536

COLORS
202
22
218
38
218
54
202
70

1600 TO
1600+11

12 colors

16
14
1536
63
1599

1536
47
1583

1536 TO 1536+

13
16
78
130
208

47

```
PHA   72
TXA   138
PHA   72
TYA   152
PHA   72

LDX   XCOUNT   174,63,6
LDA   COLORS,X 189,64,6            PLA  104
TAY   168                          TAY  168
INX   232                          PLA  104
LDA   COLORS,X 189,64,6            TAX  170
STA   WSYNC    141,10,212          PLA  104
STA   53267    141,19,208          RTI  64
STY   53272    140,24,208
INX   232                   LDA COLORS,X
CPX   #12      224,12       STA 53271
BNE   CONT     208,2        INX
LDX   #0       162,0
CONT! STX XCOUNT 142,63,6
```

POKE 1599, 0
POKE 512,0 : POKE 513,6
54286,192 or 64

189,64,6
141,23,208
232

```
GR. 2: SE. 2,0: POKE 16,64: POKE 53774,64:
DL=PEEK(560)+256*PEEK(561): POKE DL+17,23:
POKE DL+18,2 : POKE 54276,4 : SET. 0,8,4:
POS. 8,9: ?#6; "YAHTZEE": POS. 4,11:
?#6; "PRESS START TO BEGIN"
POS 9,13: ?#6; "ANGELETTE/WEISSENBERG/ANSONITE"

130 IF PEEK(53279)<>6 GOTO 130
```

```
104,169,7,162,6,160,48,32,92,228,96

X=USR(1625)
```

LDA #0
STA 1568
JMP EXVBLK — 18

1536
32
1568

4C
16776
4 69
YAHTZEE 12

Γ 17 ㄱ 5

1599
1536
63

| 124

L 26 ┘ 3

7
15~~8~~4
1536
~~88~~
48

740
1570
1580
1610
1630

1584
169,0
141,63,6
1591
76, ~~209~~,~~231~~
98, 228
POKE 546, 48
POKE 547, 6

32 Γ 98 228

SO. 0,50,12,8: F. I=1 TO 20:N. I: SO. 0,0,0,0

546
1512
34

34,2

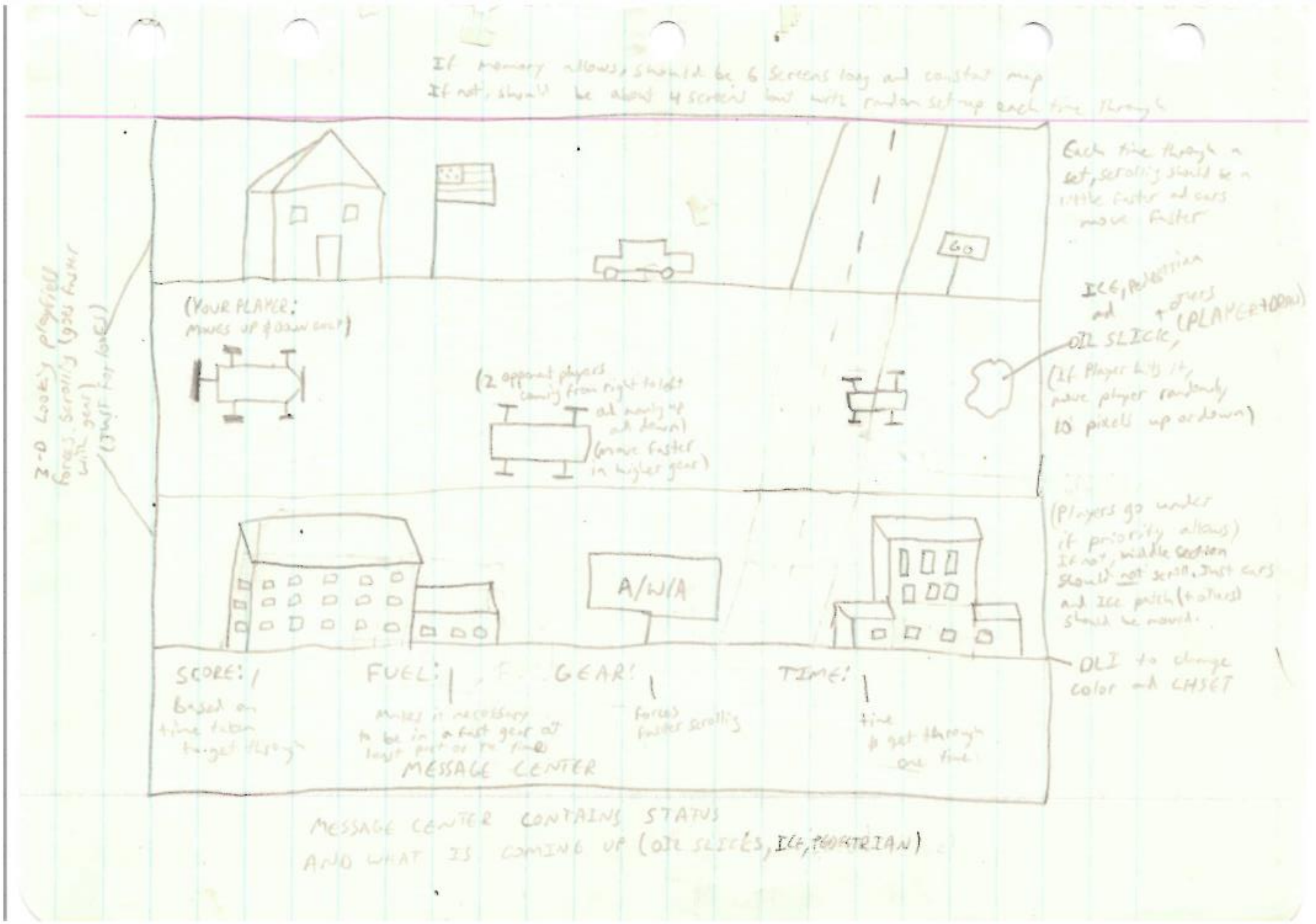
NEED V,B

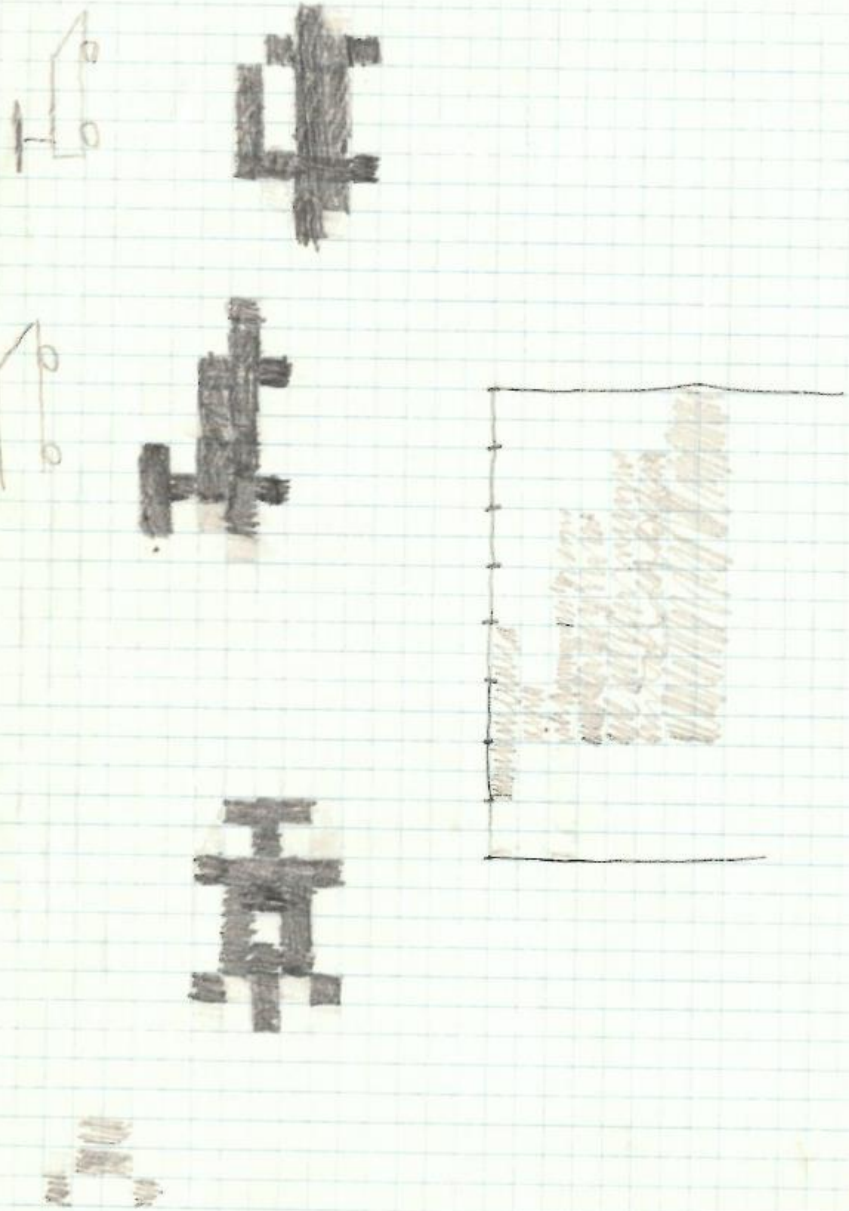
to reset 1599

~~LDS~~
7
15~~8~~4
1536
48

YAHTZEE.BAS

#UNFINISHED GAMES


If memory allows, should be 6 screens long and constant map
If not, should be about 4 screens but with random set up each time through
Each time through a set, scrolling should be a little faster and cars move faster
GO
ICE, PEDESTRIAN and OIL SLICK, + others (PLAYER+DRAW)
(If Player hits it, move player randomly 10 pixels up or down)
(YOUR PLAYER: MOVES UP & DOWN ONLY)
(2 opponent players coming from right to left and moving up and down)
(move faster in higher gear)
3-D Looking playfield forces scrolling (goes faster with gear) (just for looks)
A/W/A
(Players go under if priority allows)
If not, middle section should not scroll. Just cars and ice patch (+ others) should be moved.
DLI to change color and CHSET
SCORE:
based on time taken to get through
FUEL:
makes it necessary to be in a fast gear at least part of the time
GEAR:
forces faster scrolling
TIME:
time to get through one time
MESSAGE CENTER
MESSAGE CENTER CONTAINS STATUS AND WHAT IS COMING UP (OIL SLICKS, ICE, PEDESTRIAN)



birds
charge
chset
for wings.
Can make through
only w/ no
wings

also have another plane flying at you. Like N. Res.

3 planes
at you
Controlled by OLLS

shoot
use missles
horiz only

house 1000 pts
can only go through if you
shot wings

tables
number
to scroll

61 X256 = 15615
+111 +110 = 15726
+1111 +113 = 15729